Chapter 10
A 3-Bladed Turbine
(Example Case 4)

[NOTE: This chapter reproduces much of the information in Chapter 7 - Example Case 1. It is meant as a stand-alone tutorial for users who will be doing 3- or 4-bladed rotors, as opposed to the two-bladed rotors considered in the first three example cases. It is not supposed to be a realistic design, since most of the components are sized for the 2-bladed examples. This is very clear in the rotor's response during the simulation!]

Generally speaking, the construction of a virtual turbine in ADAMS/WT mimics the construction of a real turbine, in that the aggregate elements are first constructed separately, then placed in the correct positions and connected together. Many of these connections have been made more automatic in this version of WT. Finally, a few site- or turbine-specific "adjustments" are made before the machine is placed on-line.

To demonstrate the entire process, this section describes the construction, in ADAMS/WT, of an example 3-bladed horizontal-axis machine. The construction steps are listed below, then discussed in more detail in the following text.

10.1 Outline

NOTE: To work through this example, you should switch into the *nrel/examples/case_4* directory before starting ADAMS/View. Assuming you have set up the environment variables and *aview.pth* file correctly (see appendix I), you can then start View and load the ADAMS/WT overlay with the by reading in the command file *wt_main.cmd*. This can be done from the FILE IMPORT menu or from the command line. At this point you should be ready to begin *case_4*.

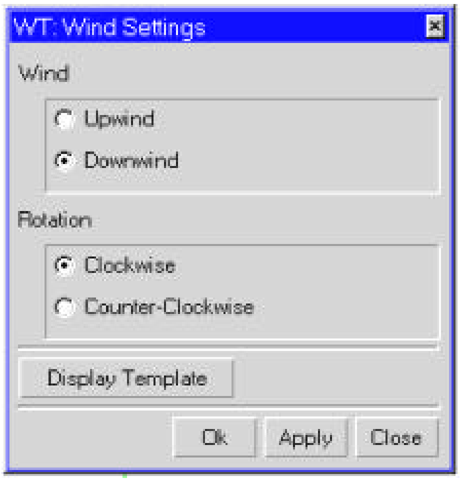
1. Set the direction of rotation and rotor orientation.
2. Create the tower aggregate element.
3. Create the nacelle aggregate element on top of tower.
4. Create the power train in the nacelle.
5. Stator
6. Low-speed Shaft
7. Motor-Generator
8. Create the 3-bladed rigid hub aggregate element.
9. Relocate the hub to the end of the low-speed shaft.
10. Create blades #1, #2 and #3.
11. Relocate blades to their attachment points on the hub.
12. Add AeroDyn aerodynamics to each blade.
13. Add gravity.
14. Add desired output requests.
15. Create a user-executable version of Solver and do the analysis.
16. Look at the results.

NOTE: In order to avoid losing your work, we recommend that you save the ADAMS/View session to a binary file after each section in the example is completed. This can be done through the FILE SAVE menu, or from the command line by typing:

file binary write file=case_4 (or just *fi bi wr fi=case_4*)

10.2 General Set-Up

This first example uses the default set-up, that is downwind and clockwise. If you wish, you can manually select Downwind and Clockwise from main WT menu using WIND/ROTATION SETUP. The internal ADAMS/View variable *dir_rot* will be set to the string "DC".

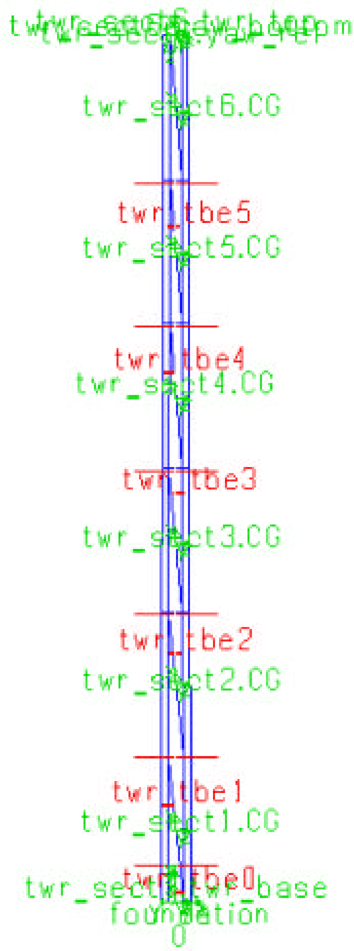


10.3 Tower

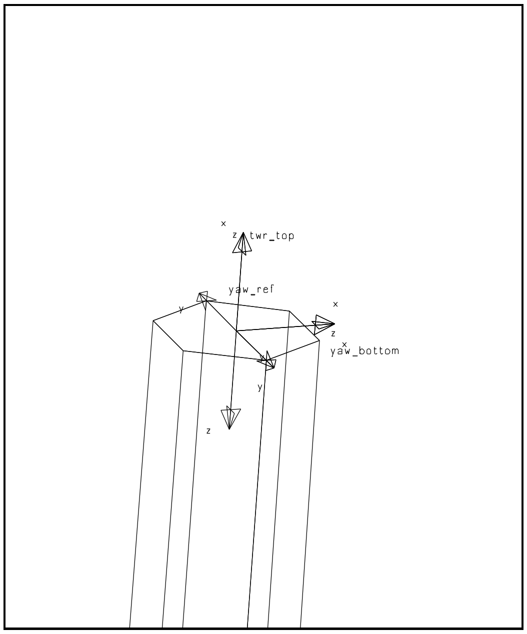
The input data for the case_4 tower are in *tower.dat* in the *examples/case_4* directory. Bring up the tower create panel and use the following values:

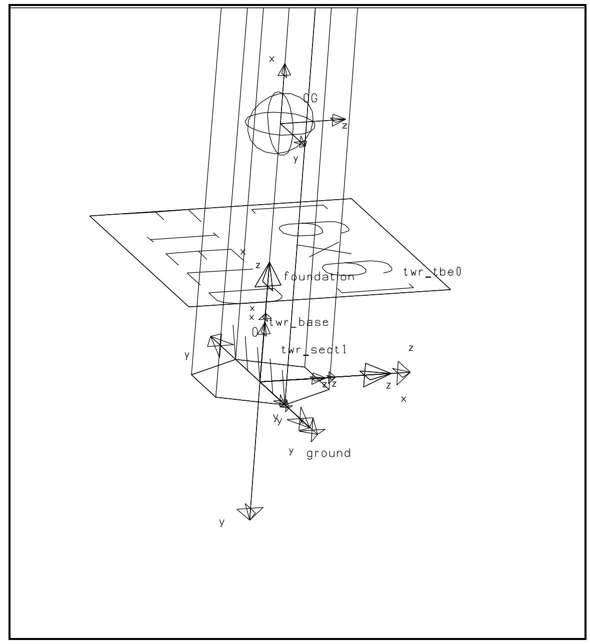
Number of Parts = 6
Tower Height = 26 m
Tower Properties File = tower.dat
Number of Sides = 6
Bottom Diameter = 1.0 m
Top Diameter = 0.8 m
Color = your choice

If everything is working properly, ADAMS/WT should display an information window which monitors the automatic tower construction. Depending on the speed of your platform, building the tower may take several minutes. When the macro terminates, the info window should disappear and you should see the tower.

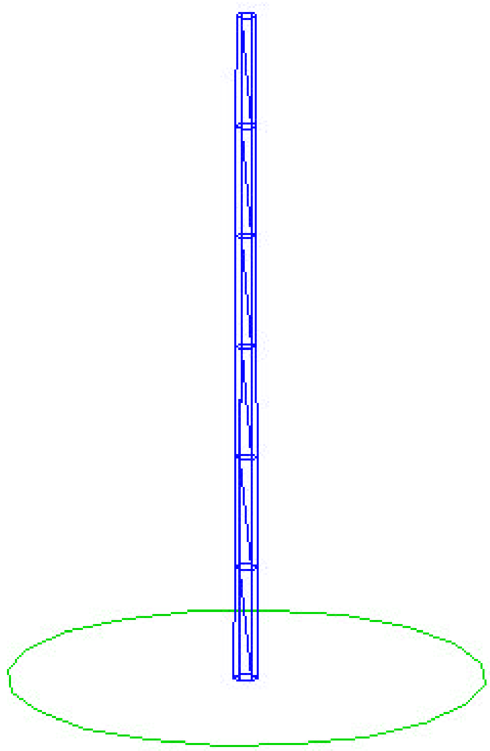


As shown on the next page, at the top of the tower is the *yaw_bottom* MARKER for later attachment to the nacelle. The bottom of the tower is a half-length tapered beam connected to the *foundation* MARKER on the ground.





If you desire, you can add some graphics to the *ground* PART to give some perspective during the subsequent modeling. Here you can see the effect of adding a 10-m radius circle graphic centered on the *O* MARKER:



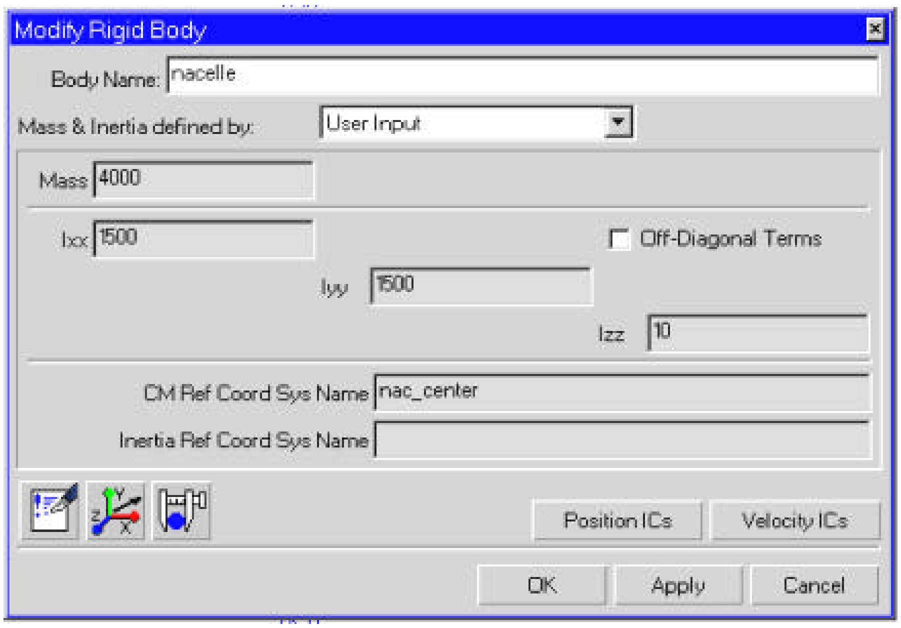
This would be a good place to save your work.

10.4 Nacelle

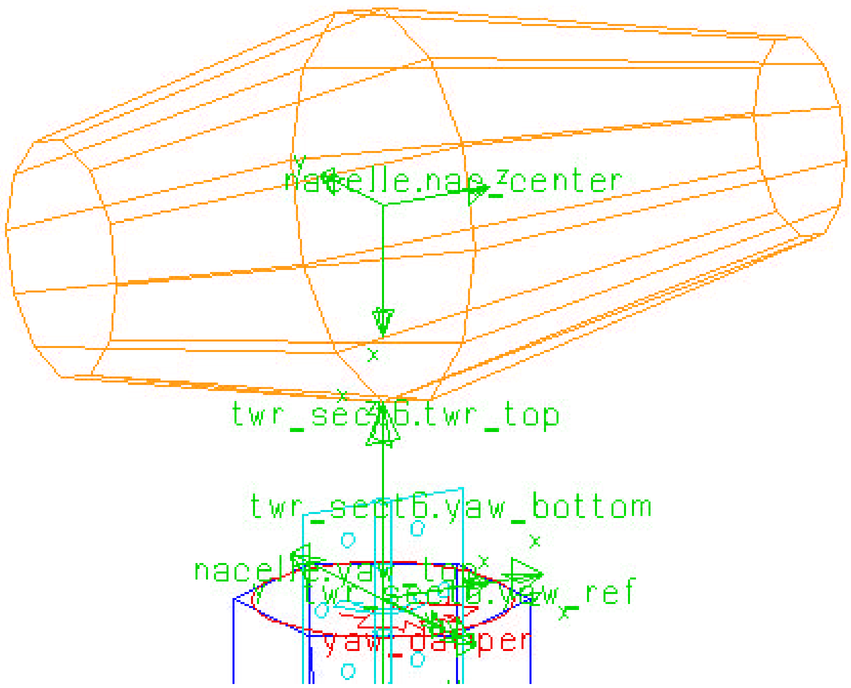
After completing the tower, we next move to the nacelle. For this example, you can use the main WT menu and enter the following data in the NACELLE CREATE panel:

Yaw Type = free_yaw
Marker_on_Tower = yaw_bottom
Shaft_Height_Above_Bearing = 1.0 m
Yaw Stiffness = 15 N-m/rad
Yaw Damping = 150 N-m-sec/rad
Diameter_at_Bearing = 1 m
Upwind Length = 1 m
Upwind Diameter = 0.6 m
Downwind Length = 1.3 m
Downwind Diameter = 0.5 m

Then, bring up the NACELLE MODIFY panel to set the mass properties for the nacelle. When you hit the MASS PROPERTIES button, it will automatically bring up the standard View PART MODIFY panel for the *nacelle* PART. You should enter the values shown in this panel, leaving the other fields blank:



After completing the PART MODIFY panel (use OK), you should Close the NACELLE MODIFY panel without doing anything else. By picking on the nacelle and rotating a bit, you should see something like this:

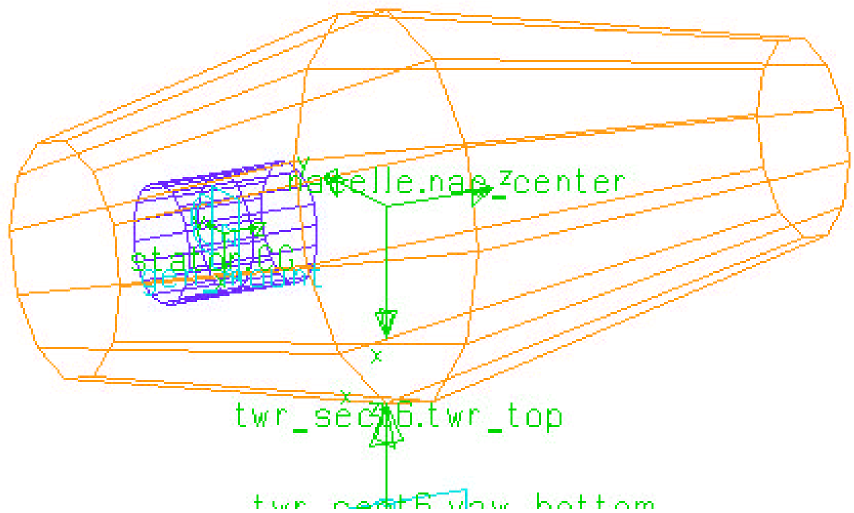


10.5 Power Train

Begin the power train by creating a generator body, called the *stator*. Remember that you can split the non-rotating inertia between the nacelle and the stator as you see fit. However, it is convenient to have a *stator* PART for connecting to one side of the motor-generator later on. Bring up the POWER_TRAIN menus (note that when you hit the POWER_TRAIN button, there will be some delay as the system retrieves the power train template), then select the GENERATOR_BODY panel and enter these values:

Location = 0,0,-0.5 m
Relative_to = nac_center
Mass = 200 kg
I_xx = I_yy = 20 kg-m^2
I_zz = 10 kg-m^2
Graphics Diameter = 0.3 m
Graphics Length = 0.4 m

This should add the *stator* to the nacelle as shown here:

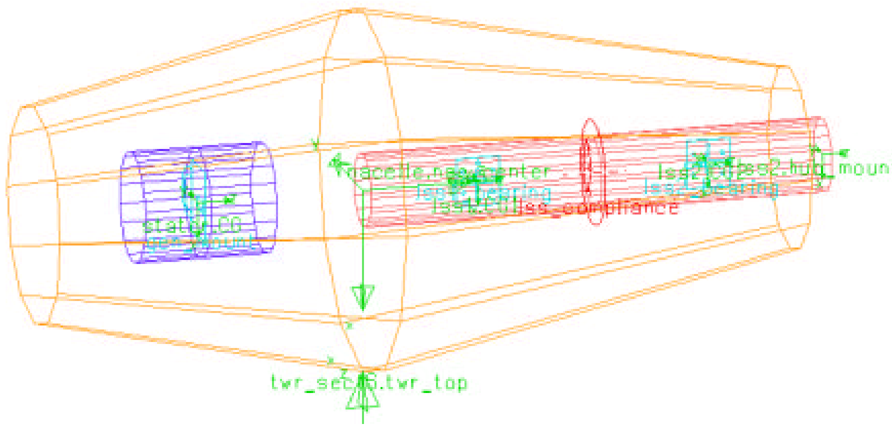


This example case does not include a high-speed shaft or gearbox. The low-speed shaft is connected directly to the motor-generator and the appropriately scaled inertia has been added to the two low-speed shaft parts. Also, this case uses a shaft with torsional flexibility only.

Select the POWER_TRAIN LOW-SPEED_SHAFT TORSION_ONLY panel and enter the following values:

Location = 0,0,0.7 m
Relative_to = nac_center
Diameter = 0.2 m
Length = 1.4 m
Stiffness = 9.826E6
Damping = 9.826E4
Mass = 100 kg
Ixx (=Iyy) = 10 kg-m^2
Izz = 1 kg-m^2

This should give you:



Since there is no high-speed shaft or gearing, next you should create the motor-generator. Here, the voltage is stepped up smoothly from zero to 240 volts over the first 200 msec of the simulation. This allows for a static solution, and for a more gentle startup. Bring up the POWER_TRAIN MOTOR-GENERATOR panel and enter these values.

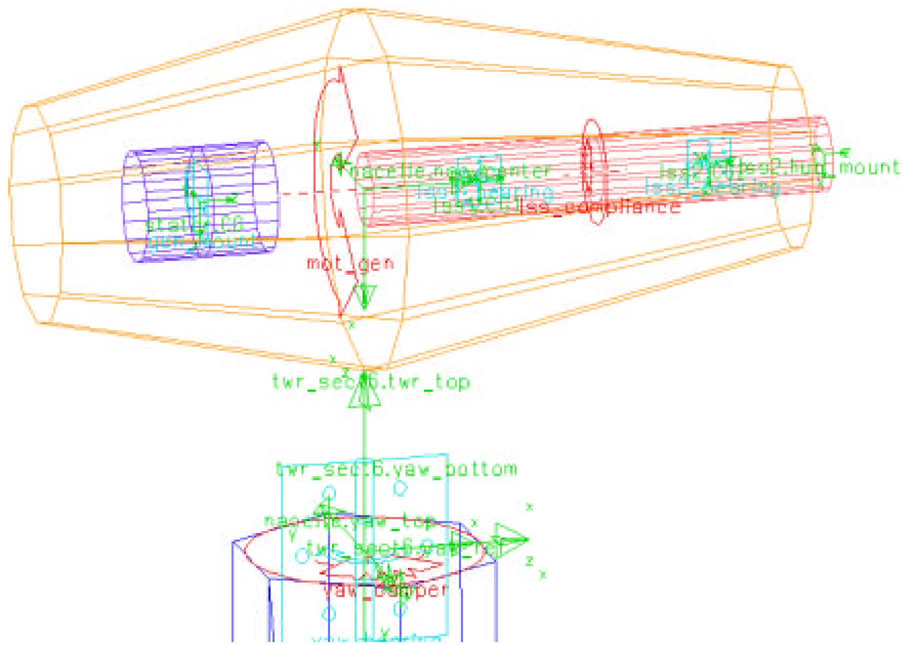
Line Voltage = STEP(TIME,0,0,0.2,240)
Desired Speed (rpm) = 60
I Marker = .hawt.lss1.CG1
J Marker = .hawt.stator.CG

The torque-voltage-speed relation in the TORQUE_FUNCTION entry is described for this case using Thevenin's equation, which is automatically generated by filling in these coefficients:

A_0 = 0.0128067
C_0 = 0.000157
C_1 = 0.00106
C_2 = 0.02428

and hitting the Generate Torque Function button to load the correct expression in the TORQUE_FUNCTION field.

Executing the motor-generator panel (OK) will create the *mot_gen* rotational SFORCE element which acts like the real motor-generator. The completed power train looks like:



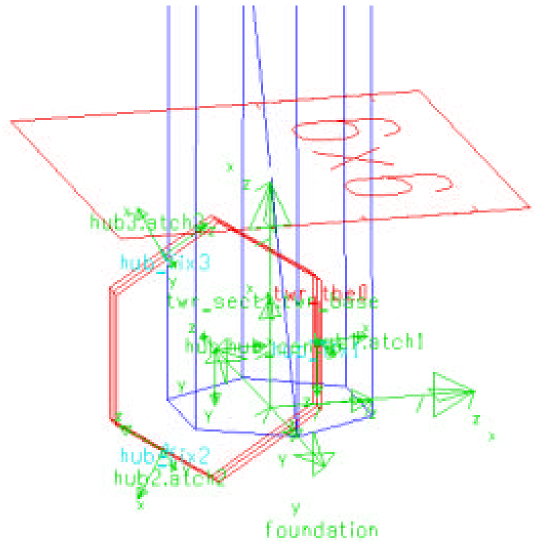
10.6 Hub

ADAMS/WT has four options for rotor hubs, 2-bladed teetering and 3-, 4- or 5-bladed rigid. For any of the hubs, you can later add flexibility between the blade attachment and the hub itself using the HUB MODIFY panels. For this example, we will create a three-bladed, rigid hub form the main WT menu using the ROTOR_HUB CREATE 3-BLADED_RIGID panel. When you select this option, WT can display a template which makes it easier to visualize exactly to what the various parameters refer.

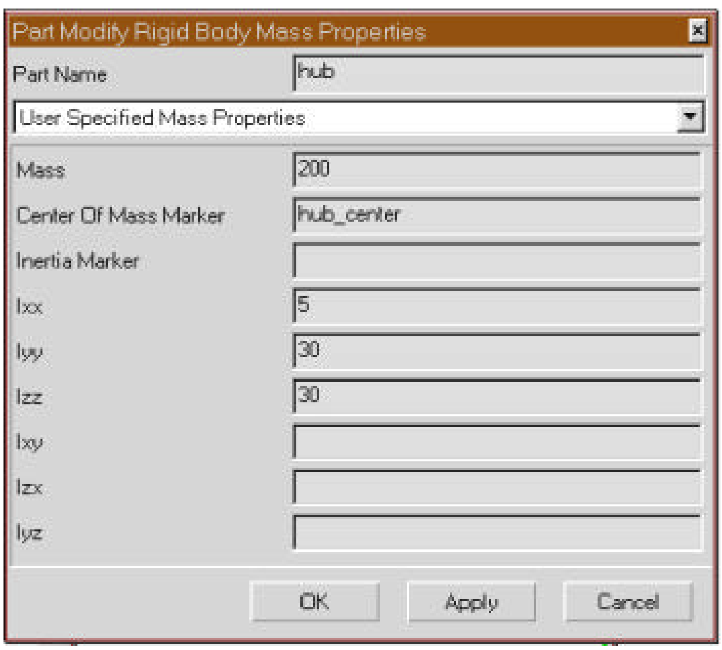
You should create a 3-bladed rigid hub using the following parameters:

Precone = 5 deg
Axial_Offset = 0.0 m
Hub_Radius = 0.5 m

This will build the *hub* PART geometry at the global origin (base of the tower). Obviously, you will later have to relocate it!

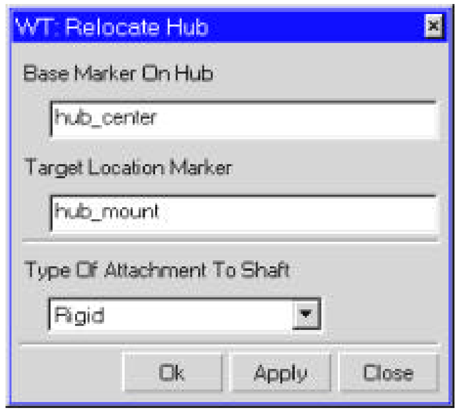


After creating the *hub* PART, you first need to bring up the ROTOR_HUB MODIFY 3-BLADED_RIGID panel, hit the MASS_PROPERTIES button and enter these data (leaving other fields blank):

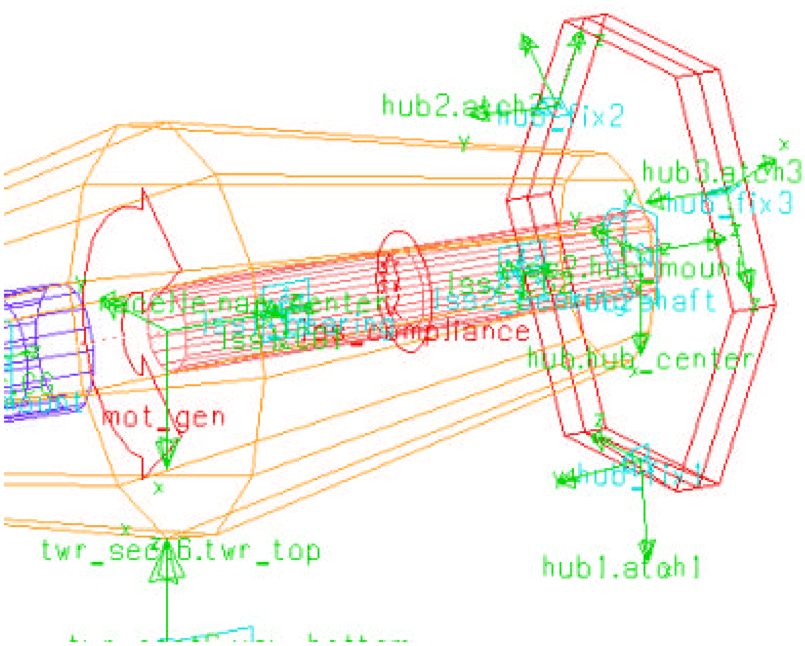


After executing this panel with OK, you will be returned to the ROTOR_HUB MODIFY 3-BLADED RIGID panel. You should simply Close out of the hub modify panel without making any other changes.

Next, you need to relocate the *hub* to the end of the low-speed shaft, using the ROTOR_HUB RELOCATE panel with the entries shown:



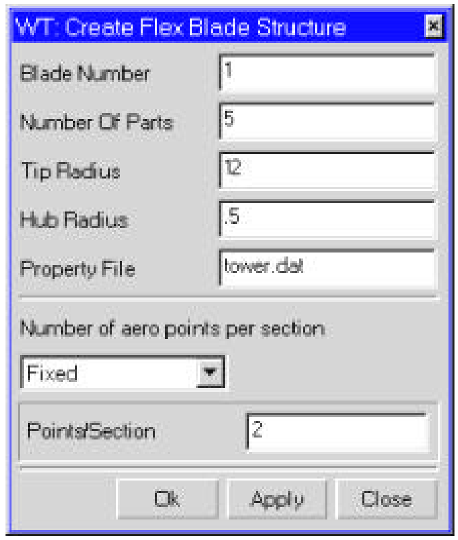
This will both move the *hub* to its correct position and create a fixed JOINT called *hub2shaft* between it and the *lss2* PART at the correct position and orientation. You should then see something like:



NOTE: This would be a good time to save your work.

10.7 Creating Rotor Blades

This example uses a fully flexible blade. The blade data for this case is found in the *blade.dat* file in the *examples/case_4* directory. Bring up the flexible blade creation panel, ROTOR_BLADE CREATE FLEXIBLE_BLADE STRUCTURAL and fill in the values shown.

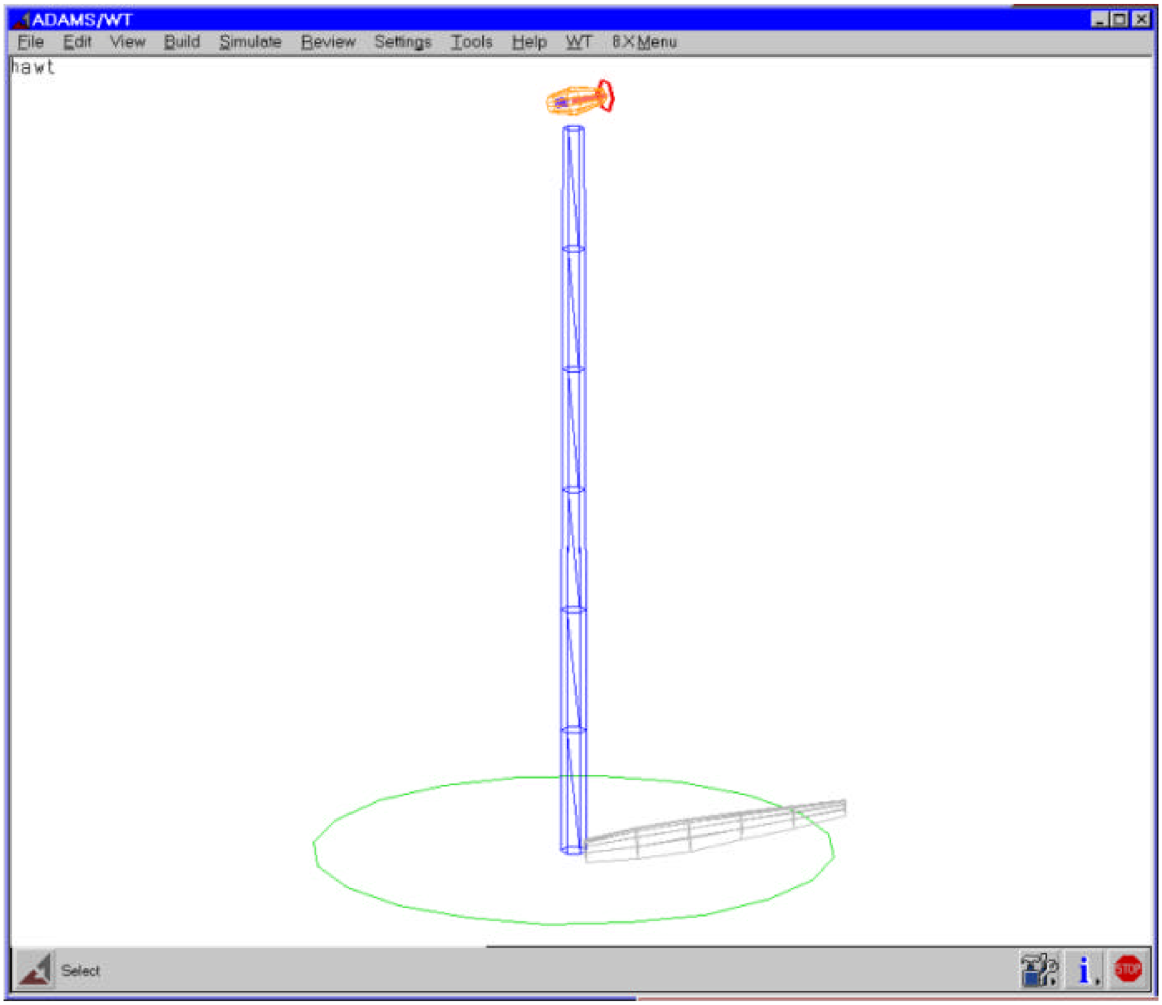


If everything is working, when you APPLY the panel, ADAMS/WT will run the auxiliary program *wtblade*.exe and should display a window which monitors the blade construction, which could take a minute or two. The blade is originally constructed on the ground. Note that the aerodynamic center locations are already in place (the *ac##* MARKERs, two per blade element) and the tip is also marked for later attachment of the tip weight. Note that we have reduced the blade radius from 13 m to 12 m for this example.

If you APPLY'ed the panel instead of OK'ing it, you can create the second blade by just switching to the Blade Number field, entering 2 instead of 1 and then selecting APPLY again. The second blade will be built exactly on top of the first. If you used OK on the first blade, you should again bring up the flexible blade creation panel through the menus, using ROTOR_BLADE CREATE FLEXIBLE_BLADE STRUCTURAL, and use the following values (WT should remember everything except the blade number.):

Blade Number = 2
Number of blade parts = 5
File of blade properties = "blade.dat"
Tip Radius =12.0 m
Tip Radius = 0.5 m
Fixed # of Aero_Points per Section = 2

Be sure to APPLY the panel this time. Then, you can create the third blade by just switching to the Blade_Number field, entering 3 instead of 2, and finally selecting OK. You will now have three blades right on top of each other at the base of the tower.



10.8 Relocating the Blades

After this, you need to relocate the blades to the correct attachment points on the hub. From the WT menu, Apply the ROTOR_BLADE RELOCATE panel with the following values. This will both move the first blade to the *atch1* MARKER on the hub and connect it to the hub with a half-length FIELD called *bl1_tbe0*.

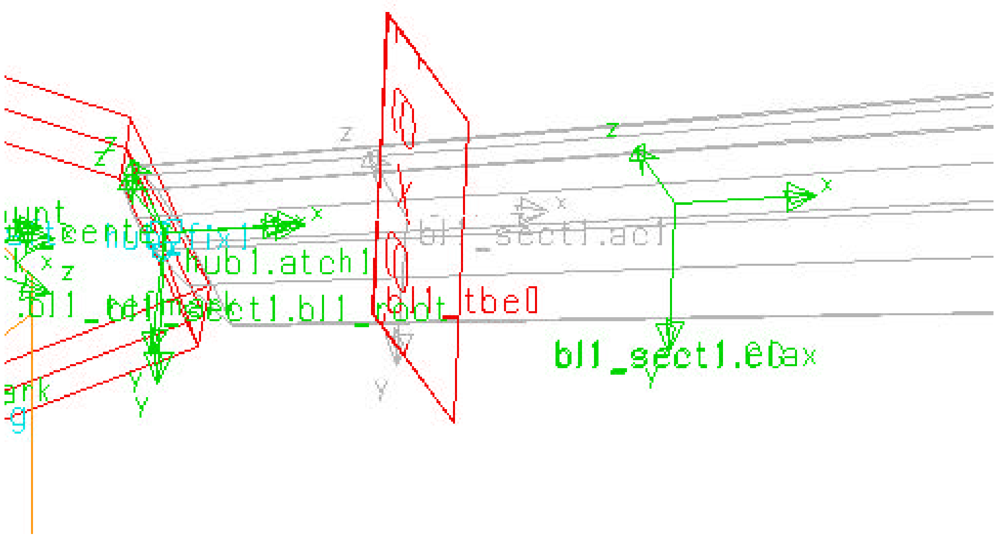
Base Marker on Blade = bl1_root
Target Location Marker = atch1
Pitch Angle = -10.8 deg

Apply the ROTOR_BLADE RELOCATE panel again with the following values to move the second blade to the *atch2* MARKER on the hub and connect it to the hub with a half-length FIELD called *bl2_tbe0*.

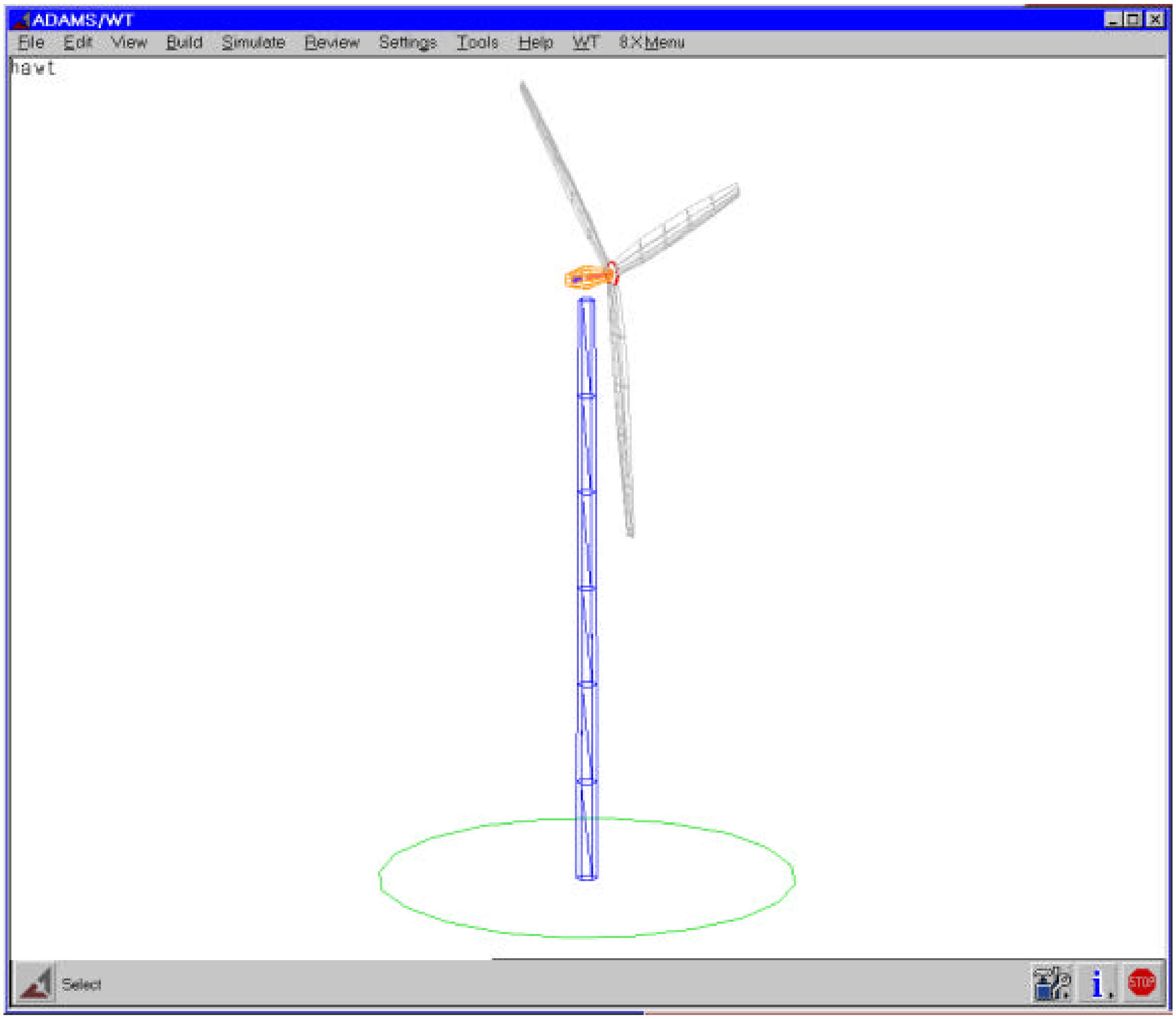
Base Marker on Blade = bl2_root
Target Location Marker = atch2
Pitch Angle = -10.8 deg

Use the ROTOR_BLADE RELOCATE panel one final time with the following values to move the third blade to the *atch3* MARKER on the hub and connect it to the hub with a half-length FIELD called *bl3_tbe0*.

Base Marker on Blade = bl3_root
Target Location Marker = atch3
Pitch Angle = -10.8 deg



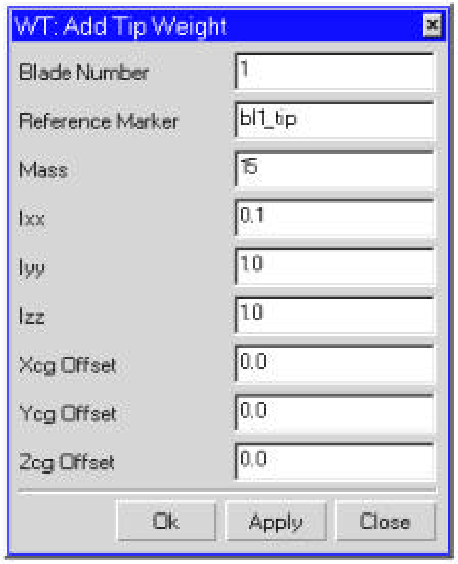
Note that, by convention, the #1 blade begins in the zero azimuth position, which is pointing straight down. The following graphic has the ADAMS/View icons turned off for clarity.



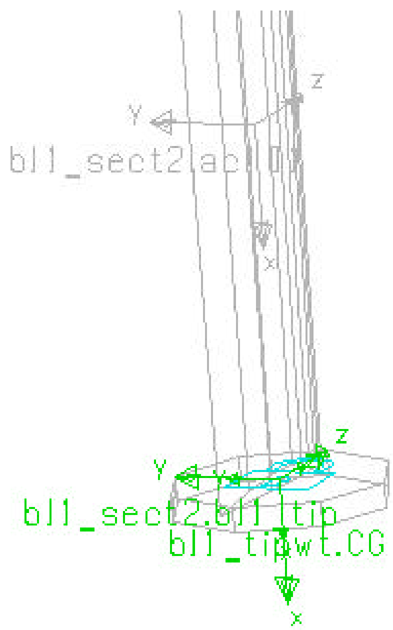
10.9 Tip Weights

This example case uses tip weights on all three blades. Tip weights are often used to give the inertial effects of an undeployed tip brake mechanism. This version of ADAMS/WT does not include deployable tip brakes as an automatically-generated aggregate element. You could, of course, add such a mechanism to each blade manually.

To add a tip weight to blade #1, bring up the ROTOR_BLADE ADD_TIP_WEIGHT panel and Apply it with the parameters shown:



This should change the blade tip to look like this:



After Apply'ing the panel for blade #1, you should change the blade number and reference marker for blade #2 and Apply again. WT should retain the values in the other fields.

Blade_number = 2
Ref_marker = bl2_tip

After Apply'ing the panel for blade #2, you should change the blade number and reference marker for blade #3 and select OK.

Blade_number = 3
Ref_marker = bl3_tip

10.10 Aerodynamics

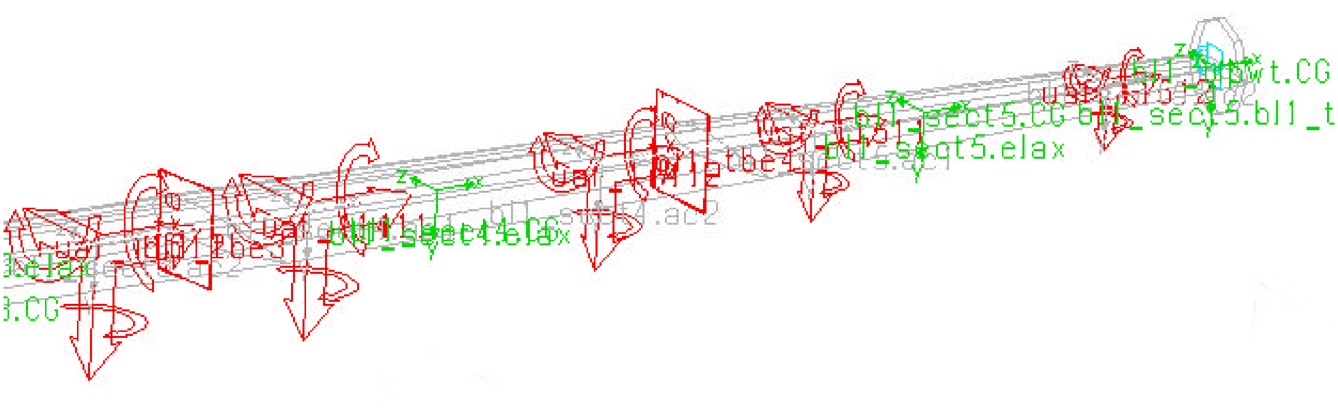
It is surprisingly simple to add aerodynamics to the model, due mainly to the large amount of up-front work done by Craig Hansen's group at the University of Utah, and the automation provided by ADAMS/WT. The AeroDyn aerodynamics subroutines are described in more detail in Appendix H. **This version of WT is designed to work with version 11.X of AeroDyn and will not work with earlier versions.**

To add the GFORCE elements which apply the aerodynamic forces computed in the AeroDyn routines to blade #1, you should bring up the AERODYNAMICS AERODYN_AERO FLEXIBLE_BLADE panel and enter the following data:

Blade_Number = 1
Number_of_Sections = 5

Apply this panel for blade #1. Then change only the Blade_Number field (to 2) and Apply the panel again for blade #2. Finally change the Blade_Number field again (to 3) and use OK to execute the panel again for blade #3.

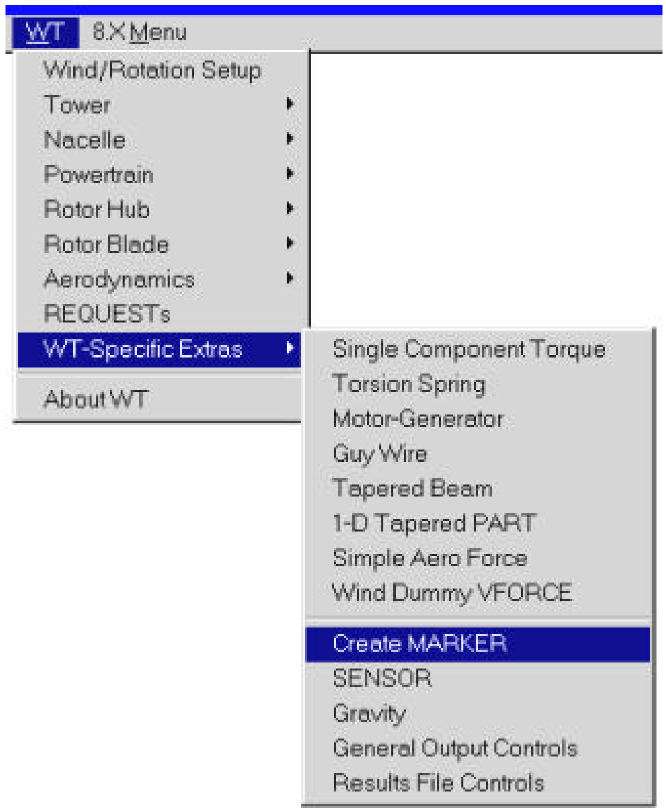
Each time you apply the panel, A/View will spend some time doing computations and, if you have the standard command window open, you should see a series of messages flash past in the dialog window, like "The floating marker FMA119110 has been created on the part .hawt.ground." After the aerodynamic GFORCE elements are added, it will be nearly impossible to make out anything on the whole model when the View icons are turned on. By itself, a section of blade #2 would look like:



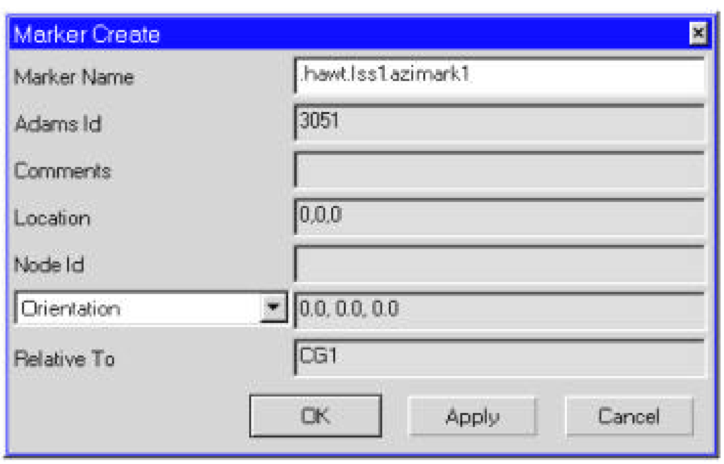
Before you can use the aerodynamics, you must add three specific MARKERs to the one of the low-speed shaft PARTs which AeroDyn will use to identify the blade azimuthal position.

These MARKERs must have the ADAMS/Solver identifiers of 305#, where # is the blade number. Also they must be aligned such that their z-axes are along the shaft axis of rotation and their x-axes point radially outward in the plane formed by the shaft and matching blade axes.

To do the first marker, you can go through the 8.X Menu to open the MARKER CREATE panel, or just use the WT-Specific Elements menu.



You should fill out the panel as shown here:



<u>Apply</u> this panel, then re-do it for the other blades:

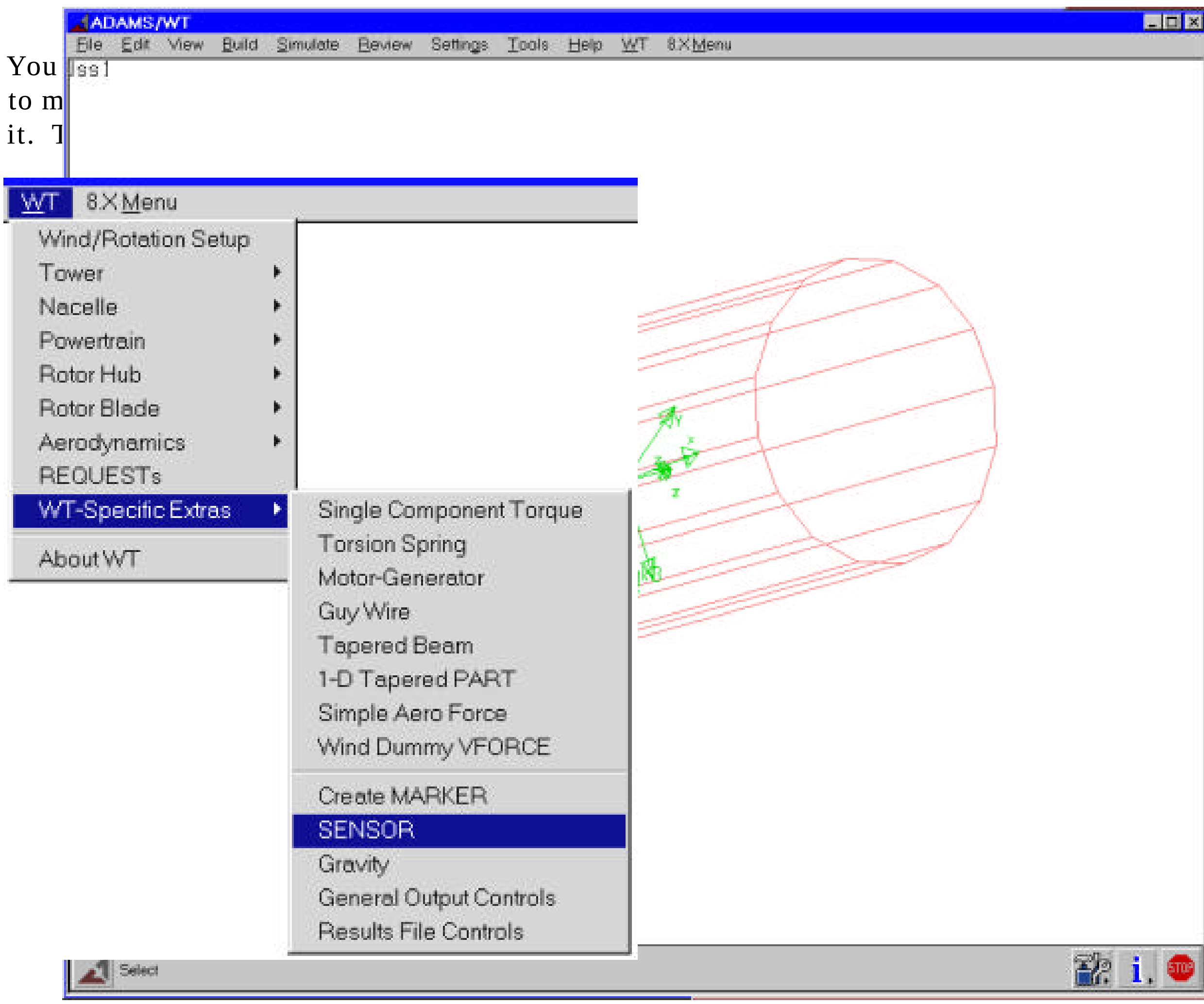
> marker_name = .hawt.lss1.azimark2
> adams_id = 3052
> location = 0,0,0
> orientation = 0,0,120
> relative_to = CG1

and

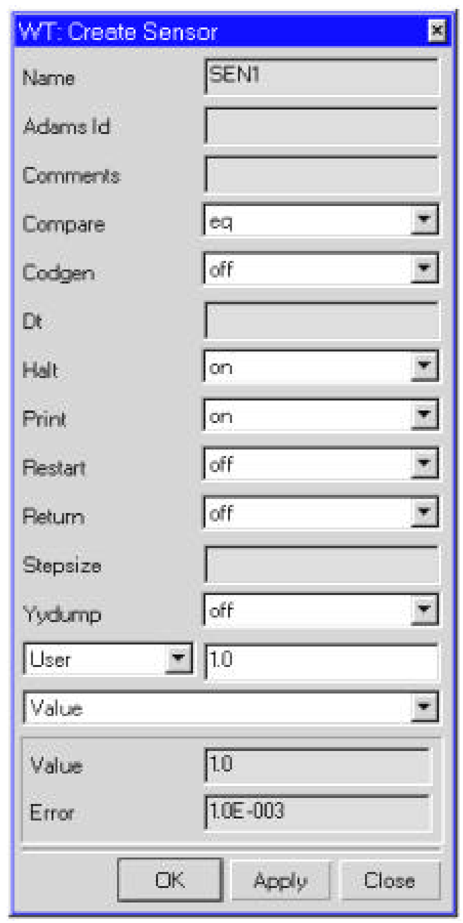
> marker_name = .hawt.lss1.azimark3
> adams_id = 3053
> location = 0,0,0
> orientation = 0,0,-120
> relative_to = CG1

After creating these three MARKERs, you can display only the *lss1* PART and if you expanded the scale of the MARKERs slightly, you would see:

You hes
to m ith
it. T



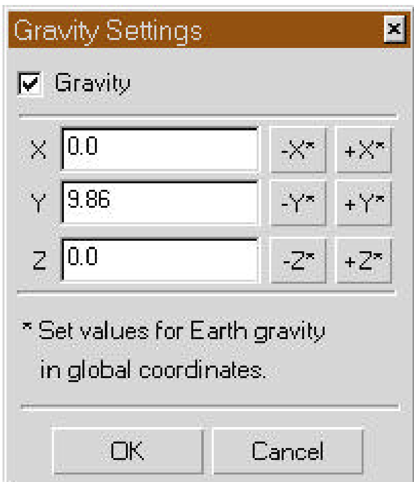
This should bring up a SENSOR creation panel with all the correct values filled in for you. You can check it against this one.



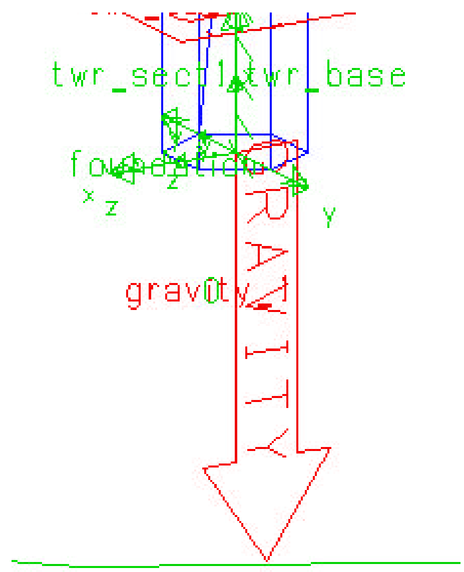
You will also need appropriate input files for AeroDyn. These are the *yawdyn.ipt*, *yawdyn.wnd* and *airfoil.dat* files which can be found in the *examples/case_4* directory. Note that *yawdyn.ipt* is not the same as for the previous examples. Finally, you must create a user-executable version of ADAMS/Solver which includes the AeroDyn routines in order to run the model with these aerodynamics. How to do this and how to run the model is covered in section 10.13 - Doing the Analysis.

10.11 Gravity

Gravity should be added to the model using the normal ADAMS/View menus (Settings / Gravity) or from the WT-Specific Extras menu, which also has a gravity choice. Complete the panel as shown:

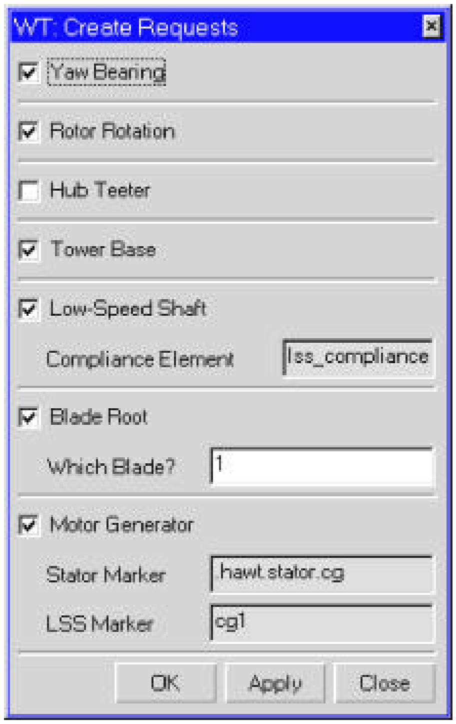


The gravity force should appear as a small arrow at the base of the tower, pointing straight down (along positive global y axis).



10.12 Output Requests

Basic requests for tabular output are now automated in ADAMS/WT version 2.0. To turn on these outputs, which will show up in the *.req* file, go back to the main WT menu and select REQUESTS to see the various different kinds of output you can solicit from ADAMS, most of which only need a confirmation to be included.



For the MOTOR-GENERATOR request, you must define the two MARKERs between which the *mot_gen* force acts. For this example, the WT defaults are correct.

M-G attachment marker on stator = .hawt.stator.CG
M-G attachment marker on LSS = .hawt.lss1.CG1

For the LOW-SPEED-SHAFT request, you must specify the *lss_compliance* torsional spring at the center of the low-speed shaft.

Which compliance element in LSS = lss_compliance

For the BLADE_ROOT request, you must specify which blade you want to monitor.

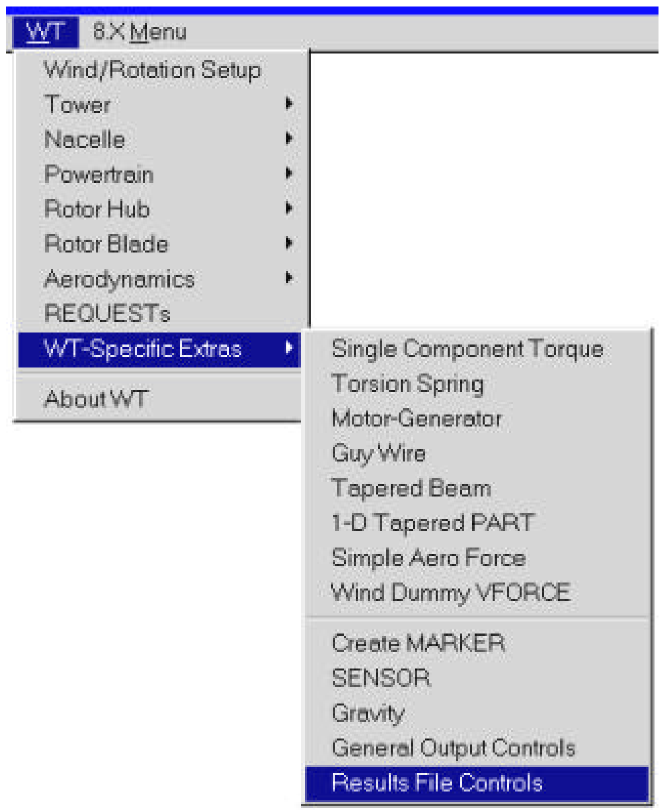
Which blade? = 1

The HUB_TEETER request, of course, is not used in this example.

You may, of course, add additional REQUESTs to the model using any and all of the methods allowed by ADAMS, i.e. standard type requests, functionally defined requests or REQSUB user-subroutine-generated requests. AeroDyn comes with an example REQSUB user-subroutine which is described in detail in the AeroDyn appendix.

As in previous versions of ADAMS/WT, to save both space and run time, we recommend that you use the standard WT-Specific Extras menu to turn off the RESULTS output. This minimizes the absolute amount of output and speeds the runs significantly. If you do need to use the results (.res) file, writing it UNFORMATTED will be both faster and produce much smaller files than the default FORMATTED output, but files will not be portable across platforms.

The Results control panel can be accessed from the WT-Specific Extras menu.

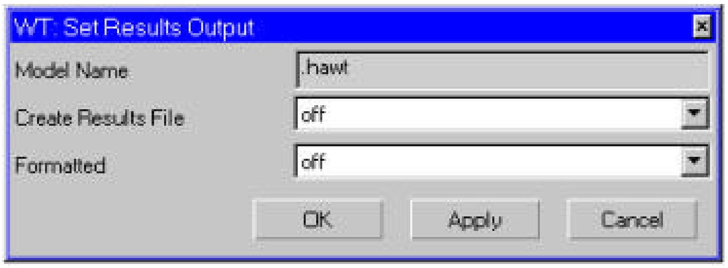


To turn off the results completely, set:

create_results_file = off
formatted = off

To keep the results file, but use the faster unformatted output, on the same panel you should set :

create_results_file = on
formatted = off



10.13 Doing the Analysis

As mentioned in above, writing output files can be a large portion of total run time, especially on platforms with small disk write caches. Generally speaking, therefore, except during debugging, it is better to use specific REQUESTs for plotting and the graphics file for animation than to postprocess from the results (*.res*) file.

The ADAMS/Solver output (*.out*) file also is normally superfluous, since the information there is a reformatted version of the request file. You can reduce the size of the *.out* file to a minimum from the main WT menu by selecting the panel for General Output Controls and setting:

print = off

You do not need to change any of the other fields in this panel. Nothing appears in your A/View model when this panel is executed; the ADAMS dataset will have a line in it which reads, "OUTPUT/NOPRINT."

In the past, it has often been necessary to "tweak" the static and dynamic solution parameters a little to get a wind turbine model to run the most smoothly. The integrators in ADAMS/Solver version 9.1, however, have been under continuous development for some time, as have the AeroDyn aerodynamics subroutines, and you should find that the combination in WT 2.0 works much better with default integrator and solver values than previously.

In addition, the numerical performance of the AeroDyn subroutines has been significantly improved. Because of this, it is now possible to run most WT-created models using the standard, default GSTIFF integrator. We recommend WSTIFF, however, because it appears to run more smoothly. Despite this appearance, GSTIFF may still run much faster. You should experiment with the integrators with your model.

To specifically select WSTIFF, from the Command Navigator, select the panel for EXECUTIVE_CONTROL SET NUMERICAL_INTEGRATOR_PARAMETERS. Cycle the Integrator Type to WSTIFF and execute the panel without changing any of the other parameters. This will put the INTEGRATOR/WSTIFF command right in the dataset and you should not need to put it in your Solver command file.

Using GSTIFF instead of the BDF integrators could make your simulations run significantly faster, possibly as much as 3-4 times faster. You may, however, see many more informational messages about integrator and solver performance "hiccups" than previously appeared using

WSTIFF. (By default, the WSTIFF integrator neglects to tell the user about most of these little problems!) As long as the integrator continues and does not get bogged down, these messages can be considered just educational. To specifically select GSTIFF, from the Command Navigator, select the panel for EXECUTIVE_CONTROL SET NUMERICAL_INTEGRATOR_PARAMETERS. Cycle the Integrator Type to GSTIFF and execute the panel without changing any of the other parameters (but be sure that interpolate=off).

At this point, your model is complete and you should change its name from the default *hawt* to *case_4* and then write it out in dataset (*.adm*) format for simulation and in View binary and command file formats for safekeeping. These actions may be accomplished from the BUILD / MODEL / RENAME and FILE / EXPORT panels, or directly from the A/View command line. In the command line window, you can type:

model modify model=hawt new=case_4
file adams write file=case_4
file command write file=case_4 entity=case_4
file binary write file=case_4

First, you should compile the AeroDyn FORTRAN code into object modules. In the *nrel/fortran* directory you should compile the files *aerosubs.f, modules.f, sensub.f* and g*fosub.f*. If you plan to later use the pre-configured AeroDyn requests instead of the standard ADAMS/WT requests, you can also include the *reqsub.f* file into the executable. (It is not needed for this example, but does not hurt anything to include the file.) The compile command, depending on your platform, should be something like

f77 -c aerosubs.f sensub.f modules.f gfosub.f (UNIX)
or
df /c /G5 /Ob2 /MD aerosubs.f modules.f sensub.f gfosub.f (NT)

Assuming that there were no errors in the compilation, you should end up with four object files, *aerosubs.o, sensub.o, modules.o* and g*fosub.o*. Note that for successful compilation, you must have the two include files *aerodyn.inc* and *bedoes.inc* in the same directory. You can then create the user-executable version of ADAMS/Solver with the menu interface step by step, or with the single long command

mdi -c cr-u i n aerosubs.o sensub.o modules.o gfosub.o -n wt20.exe exit

or for NT

mdi cr-u n aerosubs.obj modules.obj sensub.obj gfosub.obj -n wt20.exe

This should leave the file *wt20.exe* in the directory. You should copy or move this file into the *examples/case_4* directory for use with just this model. You should switch back to the *examples/case_4* directory and you will be ready to try out the *case_4* model.

Because you are running a user-executable version of ADAMS/Solver and will need special Solver commands to run it, and because you will often be running many simulations in a row, it is usually more convenient to run the code from the system command line instead of submitting it directly from the ADAMS/View. To do this you must first create an ADAMS/Solver command file *(.acf)* to control the simulation. Using your editor, create a text file named *case_4.acf* with the following contents:

```
case_4
case_4
integrator/err=.01
sim/dyn,end=0.25,step=50
integrator/err=1e-3
sim/dyn,end=0.5,step=50
sim/dyn,end=2,dtout=.02
sim/dyn,end=4,dtout=.02
sim/dyn,end=6,dtout=.02
sim/dyn,end=8,dtout=.02
sim/dyn,end=10,dtout=.02
stop
```

To run the code you can again use the menu interface step by step, or enter the single long command at the system prompt:

mdi -c ru-u i wt20.exe case_4.acf exit

or for NT

mdi ru-u wt20.exe case_4.acf exit

At this point, ADAMS/Solver should start up and the simulation progress should be displayed on screen. You can expect some difficulty with simulation startup, and a lot of warning messages about corrector convergence during the run, but these can both be ignored as long as the simulation recovers. The program log is also written to the file *case_4.msg*. When the run is complete, you should be returned to the system prompt and the simulation results should be in the files *case_4.gra* and *case_4.req*. The *.msg* file should contain something very similar to this:

```
**************************************************************
*                                                            *
*                Mechanical Dynamics, Inc.                   *
*                                                            *
*                       A D A M S                            *
*                                                            *
*      Automatic Dynamic Analysis of Mechanical Systems      *
*                                                            *
*                      Version 9.1                           *
*                                                            *
*  ADAMS/Solver, ADAMS/Android, ADAMS/Animation, ADAMS/FEA,  *
*  ADAMS/Real-Time Kinematics, ADAMS/Vehicle, ADAMS/View,    *
*      Collectively known as the ADAMS Product Line          *
*                    copyright C 1997                        *
*  By Mechanical Dynamics, Inc., Ann Arbor, Michigan U.S.A.  *
*                                                            *
**************************************************************
*                                                            *
*        Confidential and proprietary information of         *
*      Mechanical Dynamics, Inc.,  Ann Arbor, Michigan       *
*                                                            *
*    All  rights  reserved.  This code may not be copied     *
*    or reproduced  in any  form,  in part  or in whole,     *
*       without the explicit prior written permission        *
*                  of the copyright owner.                   *
*                                                            *
**************************************************************
*                                                            *
*      All product names in the ADAMS Product Line are       *
*          trademarks of Mechanical Dynamics, Inc.           *
*                                                            *
**************************************************************
```

```
*                                                            *
*                RESTRICTED RIGHTS LEGEND                    *
*                                                            *
*  If  the  Software  and  Documentation  are  provided  in  *
*  connection  with a  government contract,  then  they are  *
*  provided with  RESTRICTED RIGHTS.  Use,  duplication, or  *
*  disclosure by the  Government is subject to restrictions  *
*  as set forth in subparagraph (c)(1)(ii) of the Rights in  *
*  Technical   Data   and   Computer   Software  clause  at  *
*  252.227-7013,  as amended.   Title  to all  intellectual  *
*                property remains with MDI.                  *
*                                                            *
**************************************************************
**************************************************************
*                                                            *
*                 A D A M S / S o l v e r                    *
*                                                            *
*       13:58:07 27-DEC-98            Version 9.1            *
*                                                            *
*                                                            *
**************************************************************


OUTFOP:IN_FILENM
     ADAMS model file .. case_4.adm

OUTFOP:OUT_FILES

   Default file names for output files

     Tabular output file:
     case_4.out

     Diagnostic file    :
     case_4.msg

     Message Database file    :
     case_4.mdb

     Graphics file      :
     case_4.gra

     Request file       :
     case_4.req

     Femdata file       :
     case_4.fem

     Results file       :
     case_4.res


INVIEW:READMDL
   Input Phase - Reading in Model


INVIEW:MESSAGE91
   ***********************************************************
   ADAMS/Solver dataset Title:
   ADAMS/View model name: case_4
   ***********************************************************

INVIEW:READ_MDL
   Reading of model complete.


INBASE:DATABASE
   Input Phase - Populating Solver database


INBASE:INP_DONE
   Input Phase Complete.


MEKINP:CPUTIME
   CPU time is 0.43062 seconds.

USRMES:USER
    SENSUB called with no errors
   ID = 1

USRMES:USER
   AeroDyn Version 11.0, University of Utah
   ID = 1

USRMES:USER
   AWT-26 ADAMS model using University of Utah aerodynamics routines v10.0
   ID = 4

USRMES:USER
    Dynamic inflow theory not used in the analysis
   ID = 5
```

```
USRMES:USER
   Only 1 line in wind file, steady wind conditions used
   ID = 7

USRMES:USER
   Detected system force units of Newtons
   ID = 1

VERINP:END_INPUT
   Input and Input Check Phase complete.

GTMODE:NUMB_DOFS
   The system has 129 kinematic degrees of freedom.

GLGETL:USER_CMND
   integrator/err=.01

GLGETL:USER_CMND
   sim/dyn,end=0.25,step=50

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The FIXED coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                           Default  |Recommended|Selected
                        |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  1.00E-02
                 NTABS_ERR |  1.00E-03 |  -------  |  1.00E-02
                           |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  1.00E-05 |  1.00E-05
                 CRABS_ERR |  1.00E-06 |  1.00E-05 |  1.00E-05


ICCALC:DISPL
   Displacement initial condition analysis...

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 255
   Number of non-zero entries ...... = 1036
   Percentage of matrix non-zero ... = 1.5932
   Total space used in MD array .... = 125386


ICCALC:VELO
   Velocity initial condition analysis...

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 255
   Number of non-zero entries ...... = 1228
   Percentage of matrix non-zero ... = 1.8885
   Total space used in MD array .... = 126202


ICCALC:ACCEL
   Acceleration initial condition analysis...

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 693
   Number of non-zero entries ...... = 4066
   Percentage of matrix non-zero ... = 0.84664
   Total space used in MD array .... = 156214


SYMBLU:DISP_VELO
   Generating the Jacobian matrix for the displacements and velocities.

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............ = 255
   Number of non-zero entries ...... = 1036
   Percentage of matrix non-zero ... = 1.5932
   Total space used in MD array .... = 125514


SYMBLU:ACCELRATN
```

```
   Generating the Jacobian matrix for the accelerations and forces.

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for the Initial Conditions
   ======================================================
   Number of equations ............. = 693
   Number of non-zero entries ...... = 4066
   Percentage of matrix non-zero ... = 0.84664
   Total space used in MD array .... = 168734


SYMBLU:DYNAMICS
   Generating the Jacobian matrix for the dynamics problem.

CODGEN:JAC_STAT

   Jacobian Matrix Statistics for a Dynamic Analysis
   ======================================================
   Number of equations ............. = 981
   Number of non-zero entries ...... = 8679
   Percentage of matrix non-zero ... = 0.90184
   Total space used in MD array .... = 294464

    Simulation         Time         Cumulative    Cumulative     Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    0.00000E+00      2.50000E-04         0             0             1
    2.50000E-04      2.50000E-04         6             1             1
    2.50000E-02      1.25000E-03        75            21             1
    5.00000E-02      5.00000E-03        99            29             1
    7.50000E-02      5.00000E-03       114            34             1
    1.00000E-01      5.00000E-03       129            39             1
    1.25000E-01      5.00000E-03       144            44             1
    1.50000E-01      5.00000E-03       159            49             1
    1.75000E-01      5.00000E-03       174            54             1
    2.00000E-01      5.00000E-03       189            59             1
    2.25000E-01      5.00000E-03       204            64             1
    2.50000E-01      5.00000E-03       219            69             1

GLGETL:USER_CMND
   integrator/err=1e-3

GLGETL:USER_CMND
   sim/dyn,end=0.5,step=50

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The FIXED coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                          Default  |Recommended|Selected
                         |-----------|-----------|--------
        Integration error |           |           |
                NTREL_ERR |  1.00E-03 |  -------  |  1.00E-03
                NTABS_ERR |  1.00E-03 |  -------  |  1.00E-03
                          |           |           |
        Corrector error   |           |           |
                CRREL_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06
                CRABS_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06

    Simulation         Time         Cumulative    Cumulative     Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    2.50000E-01      5.00000E-03       219            69             1
    2.55000E-01      5.00000E-03       222            70             1
    2.75000E-01      5.00000E-03       234            74             1
    3.00000E-01      5.00000E-03       249            79             1
    3.25000E-01      5.00000E-03       264            84             1
    3.50000E-01      5.00000E-03       279            89             1
    3.75000E-01      5.00000E-03       294            94             1
    4.00000E-01      5.00000E-03       309            99             1
    4.25000E-01      5.00000E-03       324           104             1
    4.50000E-01      5.00000E-03       339           109             1
    4.75000E-01      5.00000E-03       354           114             1
    5.00000E-01      5.00000E-03       369           119             1

GLGETL:USER_CMND
   sim/dyn,end=2,dtout=.02

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The FIXED coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:
```

```
                           Default  |Recommended|Selected
                        |-----------|-----------|--------
      Integration error |           |           |
              NTREL_ERR |  1.00E-03 |  -------  |  1.00E-03
              NTABS_ERR |  1.00E-03 |  -------  |  1.00E-03
                        |           |           |
      Corrector error   |           |           |
              CRREL_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06
              CRABS_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06

 Simulation         Time         Cumulative    Cumulative     Integrator
    Time            Step         Iterations    Steps Taken      Order
 ___________      ___________    __________    ___________    __________
 5.00000E-01      5.00000E-03        369           119            1
 5.05000E-01      2.00000E-02        372           120            2
 6.60000E-01      2.00000E-02        396           128            2
 8.20000E-01      2.00000E-02        420           136            2
 9.80000E-01      2.00000E-02        445           144            2
 1.14000E+00      2.00000E-02        469           152            2
 1.30000E+00      2.00000E-02        494           160            3
 1.46000E+00      2.00000E-02        516           168            3
 1.62000E+00      2.00000E-02        539           176            3
 1.78000E+00      2.00000E-02        557           184            3
 1.94000E+00      2.00000E-02        579           192            3

GLGETL:USER_CMND
   sim/dyn,end=4,dtout=.02

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The FIXED coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                           Default  |Recommended|Selected
                        |-----------|-----------|--------
      Integration error |           |           |
              NTREL_ERR |  1.00E-03 |  -------  |  1.00E-03
              NTABS_ERR |  1.00E-03 |  -------  |  1.00E-03
                        |           |           |
      Corrector error   |           |           |
              CRREL_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06
              CRABS_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06

 Simulation         Time         Cumulative    Cumulative     Integrator
    Time            Step         Iterations    Steps Taken      Order
 ___________      ___________    __________    ___________    __________
 2.00000E+00      2.00000E-02        588           195            3
 2.02000E+00      2.00000E-02        591           196            3
 2.20000E+00      2.00000E-02        616           205            3
 2.40000E+00      2.00000E-02        645           215            3
 2.60000E+00      2.00000E-02        675           225            3
 2.80000E+00      2.00000E-02        740           245            3
 3.00000E+00      2.00000E-02        770           255            3
 3.20000E+00      2.00000E-02        800           265            3
 3.40000E+00      2.00000E-02        830           275            3
 3.60000E+00      2.00000E-02        863           285            3
 3.80000E+00      2.00000E-02        893           295            3
 4.00000E+00      2.00000E-02        923           305            3

GLGETL:USER_CMND
   sim/dyn,end=6,dtout=.02

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The FIXED coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                           Default  |Recommended|Selected
                        |-----------|-----------|--------
      Integration error |           |           |
              NTREL_ERR |  1.00E-03 |  -------  |  1.00E-03
              NTABS_ERR |  1.00E-03 |  -------  |  1.00E-03
                        |           |           |
      Corrector error   |           |           |
              CRREL_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06
              CRABS_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06

 Simulation         Time         Cumulative    Cumulative     Integrator
    Time            Step         Iterations    Steps Taken      Order
 ___________      ___________    __________    ___________    __________
 4.00000E+00      2.00000E-02        923           305            3
 4.02000E+00      2.00000E-02        926           306            3
 4.20000E+00      2.00000E-02        957           315            3
 4.40000E+00      2.00000E-02        987           325            3
 4.60000E+00      2.00000E-02       1020           335            3
```

```
    4.80000E+00      2.00000E-02       1054            345           3
    5.00000E+00      2.00000E-02       1095            355           4
    5.20000E+00      2.00000E-02       1135            365           4
    5.40000E+00      2.00000E-02       1174            375           4
    5.60000E+00      2.00000E-02       1214            385           5
    5.80000E+00      2.00000E-02       1260            395           5
    6.00000E+00      2.00000E-02       1310            405           5

GLGETL:USER_CMND
   sim/dyn,end=8,dtout=.02

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The FIXED coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                            Default  |Recommended|Selected
                         |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  1.00E-03
                 NTABS_ERR |  1.00E-03 |  -------  |  1.00E-03
                           |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06
                 CRABS_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06

    Simulation         Time         Cumulative    Cumulative     Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    6.00000E+00      2.00000E-02       1310            405           5
    6.02000E+00      2.00000E-02       1315            406           5
    6.20000E+00      1.00000E-02       1395            424           4
    6.40000E+00      2.00000E-02       1519            460           3
    6.60000E+00      2.00000E-02       1558            470           4
    6.80000E+00      2.00000E-02       1598            480           4
    7.00000E+00      2.00000E-02       1638            490           4
    7.20000E+00      2.00000E-02       1678            500           4
    7.40000E+00      2.00000E-02       1718            510           4
    7.60000E+00      2.00000E-02       1758            520           4
    7.80000E+00      2.00000E-02       1798            530           4
    8.00000E+00      2.00000E-02       1838            540           4

GLGETL:USER_CMND
   sim/dyn,end=10,dtout=.02

DBANNR:BDF
   Begin the dynamic analysis.

   The system is modelled with DAEs.
   The FIXED coefficient BDF method will be used.

DBANNR:BDF_TABLE
   The operating values of the error tolerances for BDF are:

                            Default  |Recommended|Selected
                         |-----------|-----------|--------
         Integration error |           |           |
                 NTREL_ERR |  1.00E-03 |  -------  |  1.00E-03
                 NTABS_ERR |  1.00E-03 |  -------  |  1.00E-03
                           |           |           |
         Corrector error   |           |           |
                 CRREL_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06
                 CRABS_ERR |  1.00E-06 |  1.00E-06 |  1.00E-06

    Simulation         Time         Cumulative    Cumulative     Integrator
       Time            Step         Iterations    Steps Taken      Order
    ___________      ___________    __________    ___________    __________
    8.00000E+00      2.00000E-02       1838            540           4
    8.02000E+00      2.00000E-02       1842            541           4
    8.20000E+00      2.00000E-02       1878            550           4
    8.40000E+00      2.00000E-02       1918            560           4
    8.60000E+00      2.00000E-02       1958            570           4
    8.80000E+00      2.00000E-02       1998            580           4
    9.00000E+00      2.00000E-02       2039            590           4
    9.20000E+00      2.00000E-02       2079            600           4
    9.40000E+00      2.00000E-02       2153            620           3
    9.60000E+00      2.00000E-02       2191            630           3
    9.80000E+00      2.00000E-02       2231            640           4
    1.00000E+01      2.00000E-02       2271            650           4

GLGETL:USER_CMND
   stop

TERM0:EXE_TERM

      ADAMS/Solver execution terminated by subprogram A3TERM


TERM0:CP_TIME
```

```
CPU time used = 56.491 seconds
```

10.14 Static Solution Note

Getting a valid static equilibrium solution is surprisingly difficult for many rotor models, including this example case. If you really need an equilibrium solution, you should first try with the default EQUILIBRIUM statement parameters. If this fails, you should try reducing the alimit parameter on the EQUILIBRIUM statement to about 5 degrees and the tlimit value to about 10 meters, while also increasing the stability parameter to about .01 and increasing the maxit value to 100. Another approach is to turn off gravity, then do a series of static solutions while slowly increasing the gravity load each time. After you get a solution, you can "play" these parameters to get optimal convergence. Note that with full gravity, no line voltage and no wind at startup, you may not get any reasonable static solution for some rotors, even though such a solution clearly does exist.

10.15 Visualizing the Results

At this point, you are ready to read the results of the simulation back into ADAMS/View to look at the responses. Switch back to the A/View window and either use the FILE menus or enter at the View command line:

file analysis read file=case_4 model=case_4

It will take View a few moments to read in the data from the graphics (*case_4.gra*) and request (*case_4.req*) files. You can then animate the results and see how the rotor responded. There are quite a few ways to animate response in View. The simplest way in the WT interface is to bring up the control panel and just hit the ANIMATE button.

10.16 Plotting Output

ADAMS/View 9.1 has a completely new plotting interface, including a large number of plotting features which can be accessed in many ways. Quick plots of request data can be made by easily made using the Plot Builder. The data can also be "surfed" this way.

For repetitive plotting of specific requests from multiple simulations, it is often best to create a View command file (*.cmd*) containing the necessary commands to create and customize all the plots for a particular run. This command file contains the same commands you could execute via the plot builder or type in at the View command line to create the plots, but is easily modifiable using a text editor for customization and changes. An example of such a command file is found in the file *plotemup.cmd* in the *examples/case_4* directory. The contents are repeated here:

```
! View command file to plot results from example case 4
! Created by A. Elliott, MDI, December '98.

xy_plot template create plot=tbl1 title="Tower Base Loads (forces)"&
 subtitle="Example Case_4" vlabel="Newtons" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=tbl1 legend=yes &
 vaxis=twr_base_loads/X,twr_base_loads/Y,twr_base_loads/Z
xy att plot_name = .tbl1 graph_area = 20, 5, 140, 85
xy curve mod curve=.tbl1.curve   legend="X"
```

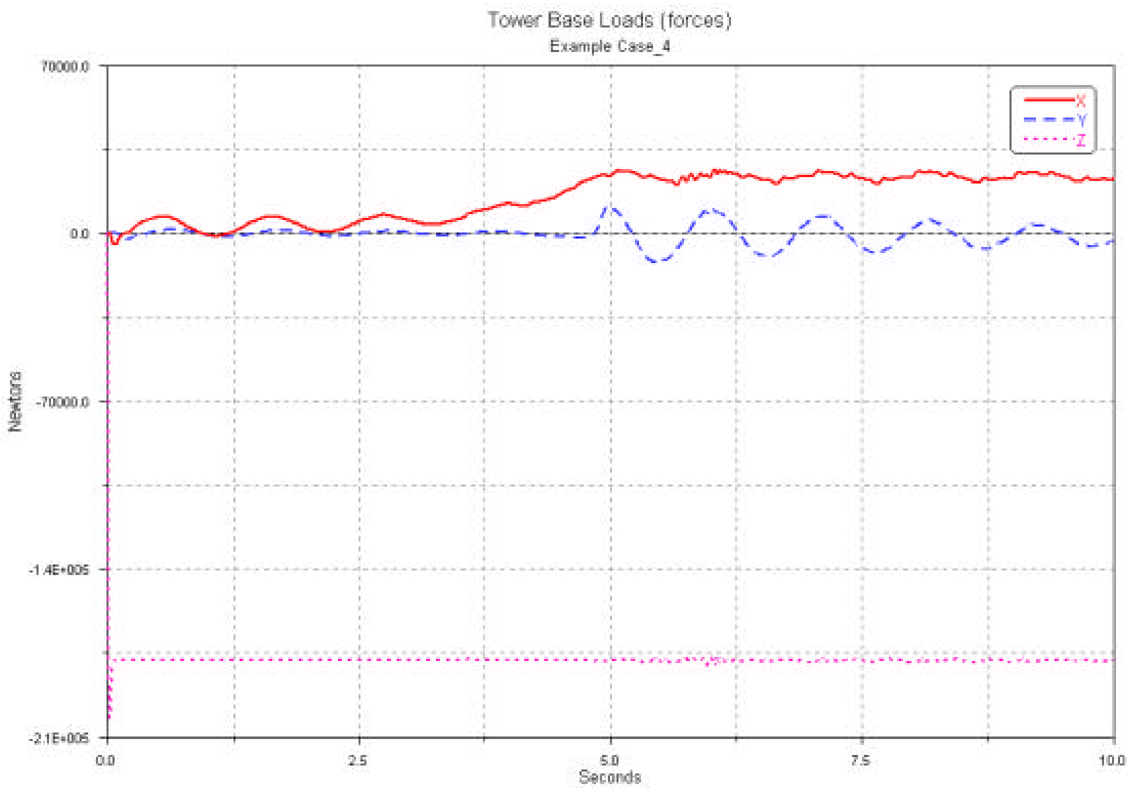
```
xy curve mod curve=.tbl1.curve_2 legend="Y"
xy curve mod curve=.tbl1.curve_3 legend="Z"
note att note=.tbl1.curve.legend    point_size = 8
note att note=.tbl1.curve_2.legend  point_size = 8
note att note=.tbl1.curve_3.legend  point_size = 8

xy_plot template create plot=tbl2 title="Tower Base Loads (torques)"&
 subtitle="Example Case_4" vlabel="Newton-Meters" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=tbl2 legend=yes &
 vaxis=twr_base_loads/R1,twr_base_loads/R2,twr_base_loads/R3
xy att plot_name = .tbl2 graph_area = 20, 5, 140, 85
xy curve mod curve=.tbl2.curve   legend="X"
xy curve mod curve=.tbl2.curve_2 legend="Y"
xy curve mod curve=.tbl2.curve_3 legend="Z"
note att note=.tbl2.curve.legend    point_size = 8
note att note=.tbl2.curve_2.legend  point_size = 8
note att note=.tbl2.curve_3.legend  point_size = 8

xy_plot template create plot=yaw1 title="Yaw Table Loads (forces)"&
 subtitle="Example Case_4" vlabel="Newtons" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=yaw1 legend=yes &
 vaxis=yaw_table_loads/X,yaw_table_loads/Y,yaw_table_loads/Z
xy att plot_name = .yaw1 graph_area = 20, 5, 140, 85
xy curve mod curve=.yaw1.curve   legend="X"
xy curve mod curve=.yaw1.curve_2 legend="Y"
xy curve mod curve=.yaw1.curve_3 legend="Z"
note att note=.yaw1.curve.legend    point_size = 8
note att note=.yaw1.curve_2.legend  point_size = 8
note att note=.yaw1.curve_3.legend  point_size = 8


xy_plot template create plot=yaw2 title="Yaw Table Loads (torques)"&
 subtitle="Example Case_4" vlabel="Newton-Meters" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=yaw2 legend=yes &
 vaxis=yaw_table_loads/R1,yaw_table_loads/R2,yaw_table_loads/R3
xy att plot_name = .yaw2 graph_area = 20, 5, 140, 85
xy curve mod curve=.yaw2.curve   legend="X"
xy curve mod curve=.yaw2.curve_2 legend="Y"
xy curve mod curve=.yaw2.curve_3 legend="Z"
note att note=.yaw2.curve.legend    point_size = 8
note att note=.yaw2.curve_2.legend  point_size = 8
note att note=.yaw2.curve_3.legend  point_size = 8

xy_plot template create plot=rpm  title="Rotor Speed"&
 subtitle="Example Case_4" vlabel="RPM" hlabel="Seconds" legend=no &
 auto=no
xy_plot curve create plot=rpm vaxis=rotor_data/Z
xy att plot_name = .rpm graph_area = 20, 5, 140, 85

xy_plot template create plot=yaw  title="Nacelle Yaw"&
 subtitle="Example Case_4" vlabel="Degrees" hlabel="Seconds" legend=no &
 auto=no
xy_plot curve create plot=yaw vaxis=rotor_data/R1
xy att plot_name = .yaw graph_area = 20, 5, 140, 85

xy_plot template create plot=root1 title="Blade Root Loads (forces)"&
 subtitle="Example Case_4" vlabel="Newtons" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=root1 legend=yes &
 vaxis=root_loads/X,root_loads/Y,root_loads/Z
xy att plot_name = .root1 graph_area = 20, 5, 140, 85
xy curve mod curve=.root1.curve   legend="X"
xy curve mod curve=.root1.curve_2 legend="Y"
xy curve mod curve=.root1.curve_3 legend="Z"
note att note=.root1.curve.legend    point_size = 8
note att note=.root1.curve_2.legend  point_size = 8
note att note=.root1.curve_3.legend  point_size = 8

xy_plot template create plot=root2 title="Blade Root Loads (torques)"&
 subtitle="Example Case_4" vlabel="Newton-Meters" hlabel="Seconds" &
 auto=no
xy_plot curve create plot=root2 legend=yes &
```

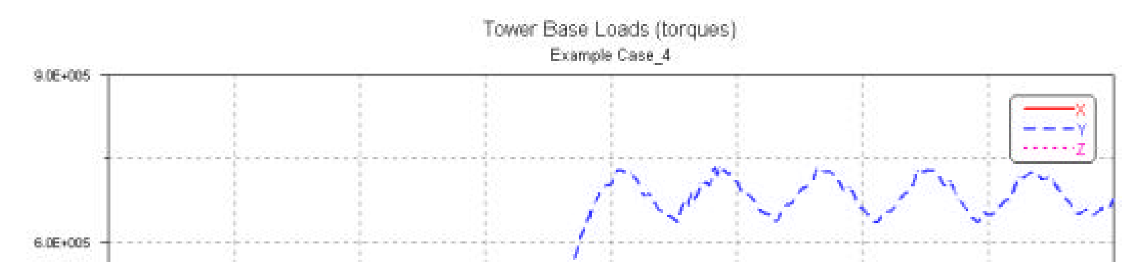
```
 vaxis=root_loads/R1,root_loads/R2,root_loads/R3
xy att plot_name = .root2 graph_area = 20, 5, 140, 85
xy curve mod curve=.root2.curve   legend="X"
xy curve mod curve=.root2.curve_2 legend="Y"
xy curve mod curve=.root2.curve_3 legend="Z"
note att note=.root2.curve.legend    point_size = 8
note att note=.root2.curve_2.legend  point_size = 8
note att note=.root2.curve_3.legend  point_size = 8

xy_plot template create plot=torque title="Motor-Generator Torque"&
 subtitle="Example Case_4" vlabel="Newton-Meters" hlabel="Seconds" legend=no &
 auto=no
xy_plot curve create plot=torque vaxis=motor_generator/X
xy att plot_name = .torque graph_area = 20, 5, 140, 85

xy_plot template create plot=twist title="Low-Speed Shaft Twist"&
 subtitle="Example Case_4" vlabel="Degrees" hlabel="Seconds" legend=no &
 auto=no
xy_plot curve create plot=twist vaxis=lss_displ/R1
xy att plot_name = .twist graph_area = 20, 5, 140, 85
```

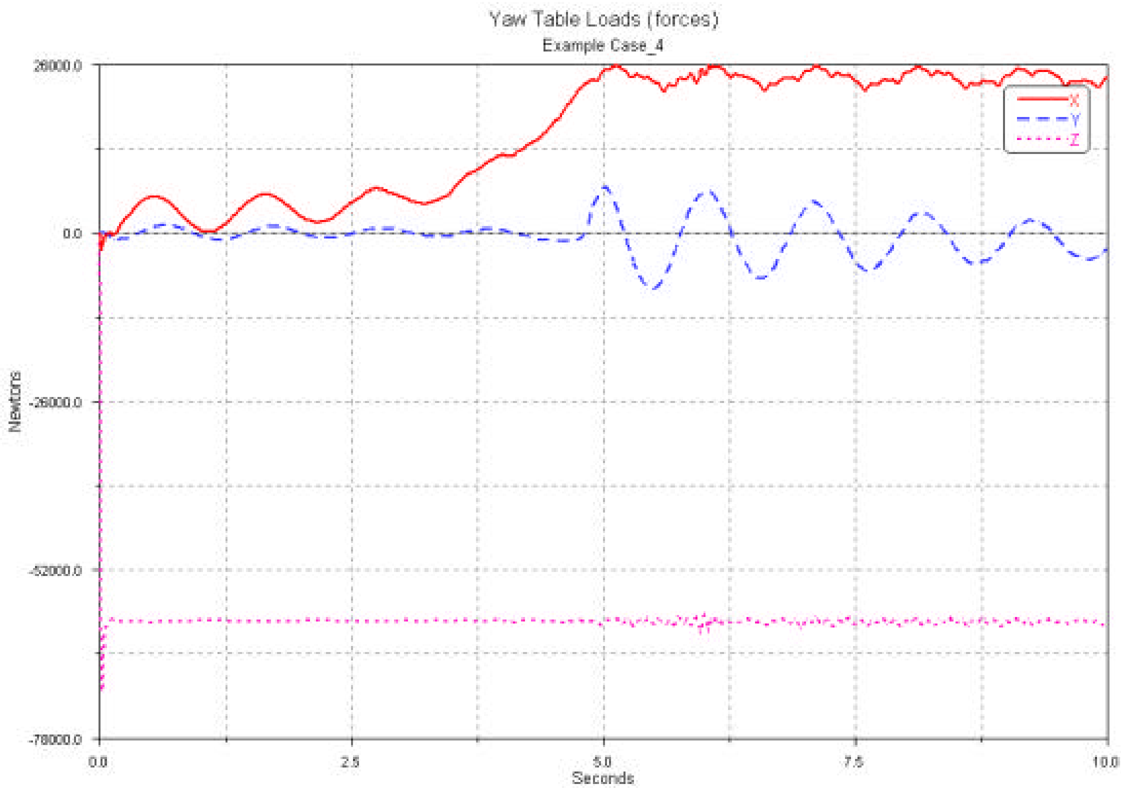
You can read in and run this command file through the FILE IMPORT menu or by entering at the View command line:

file command read file=plotemup

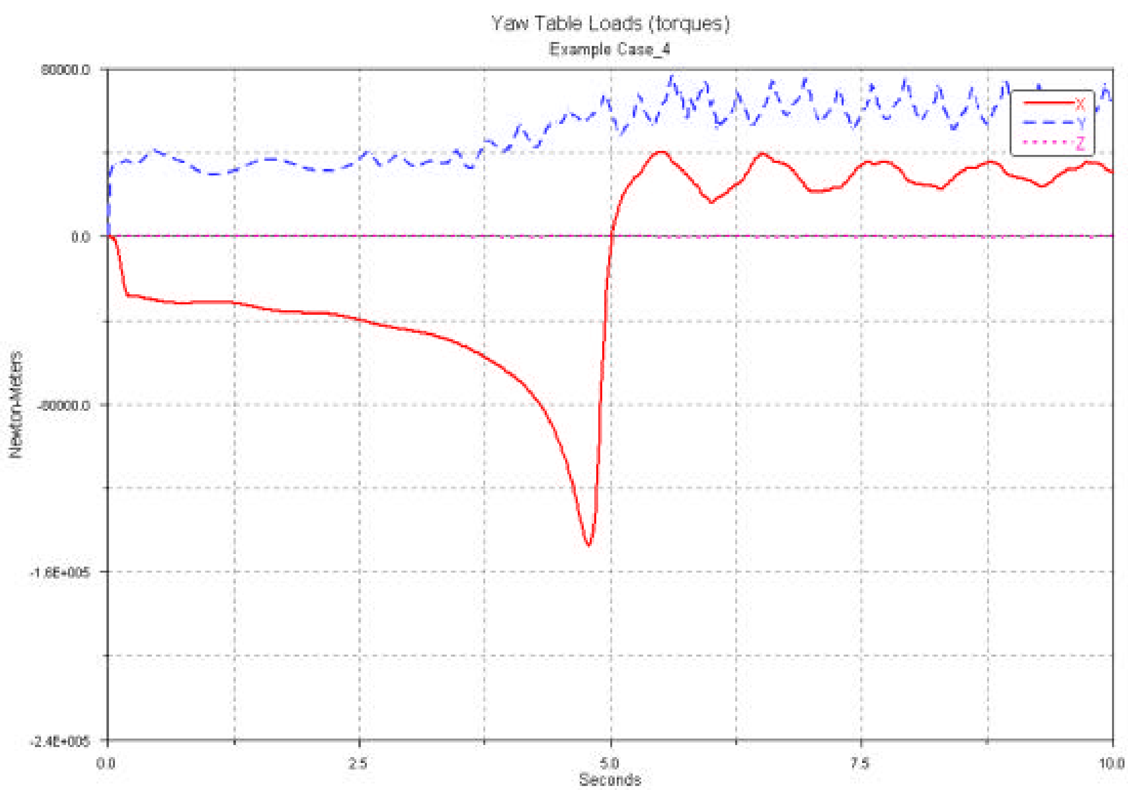
The example plots below can be used to confirm that your model and WT executable are working correctly.



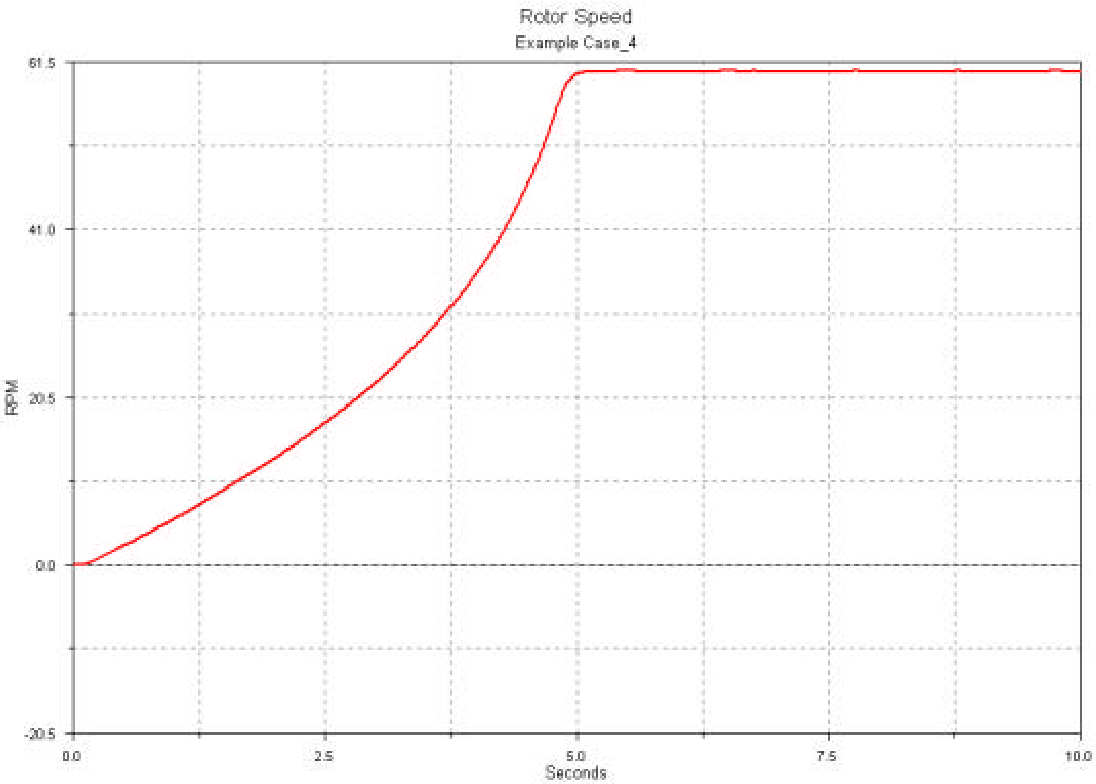


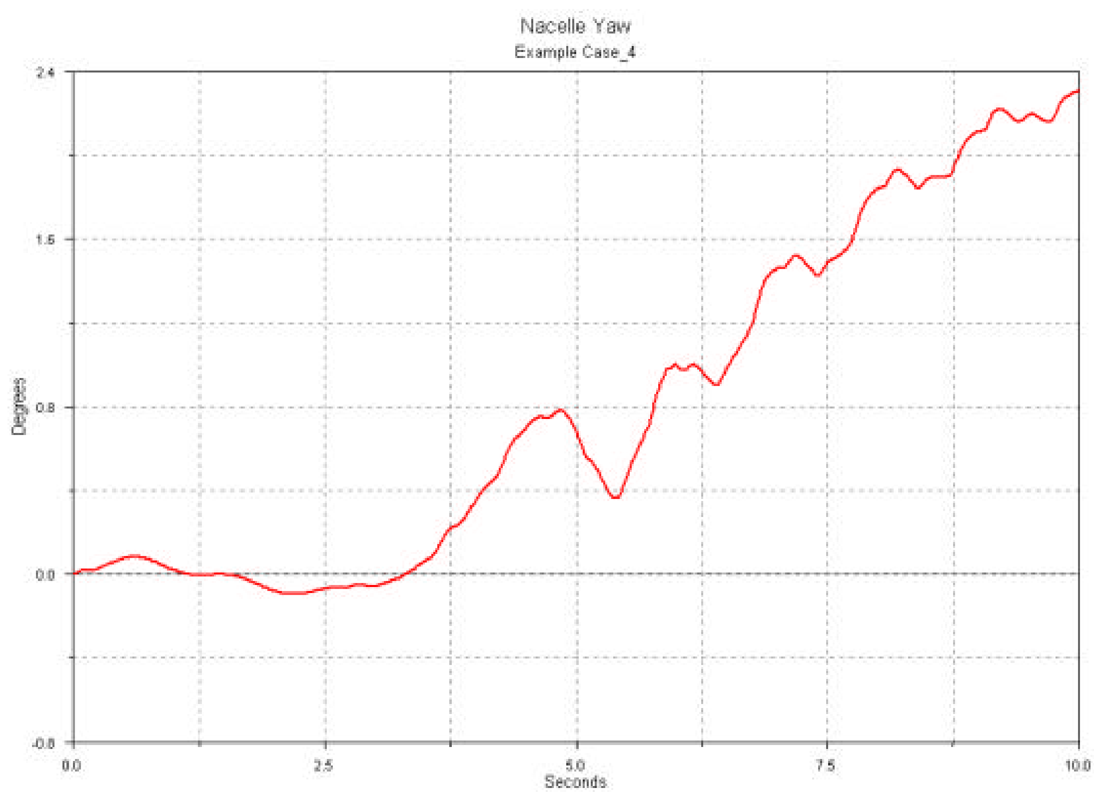


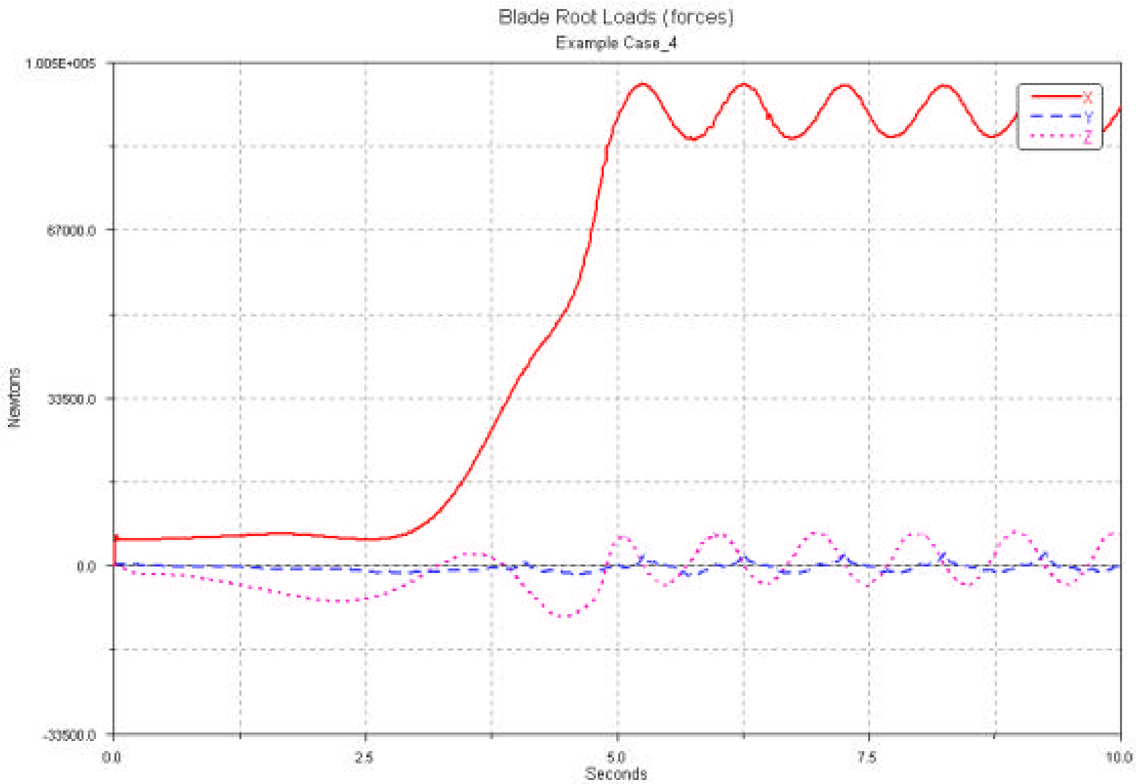
Yaw Table Loads (forces)
Example Case_4
26000.0
0.0
-26000.0
-52000.0
-78000.0
Newtons
0.0
2.5
5.0
7.5
10.0
Seconds
X
Y
Z

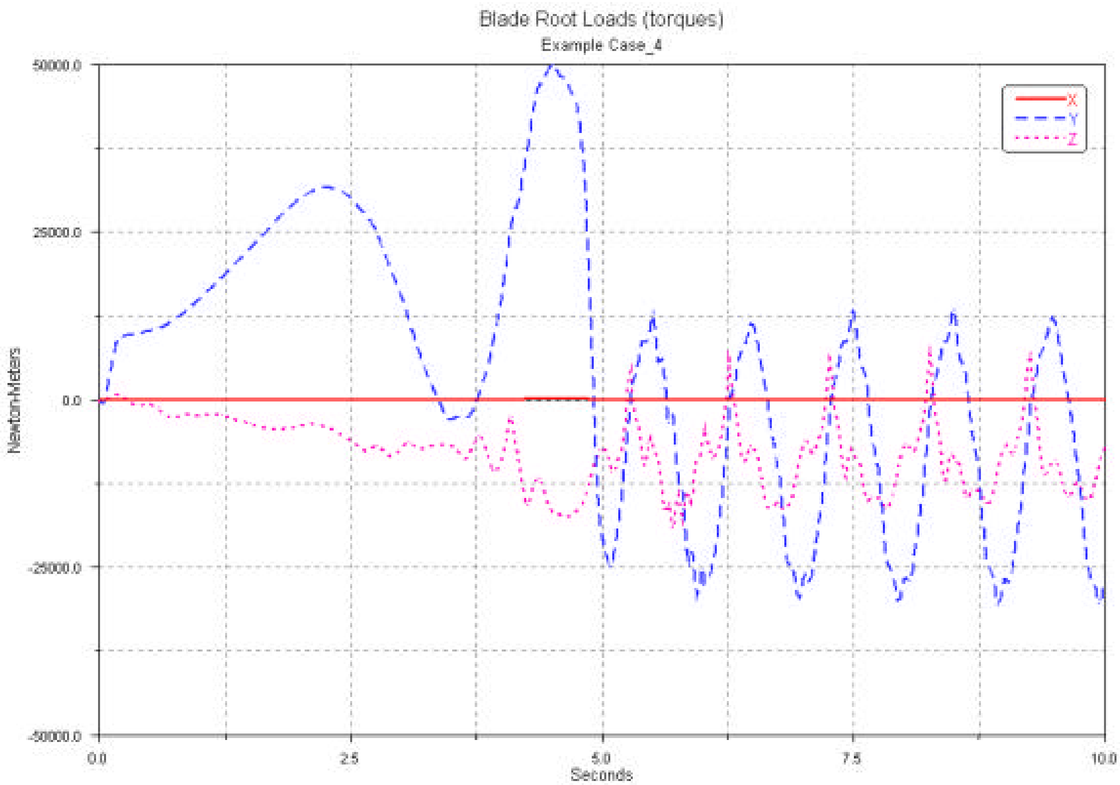


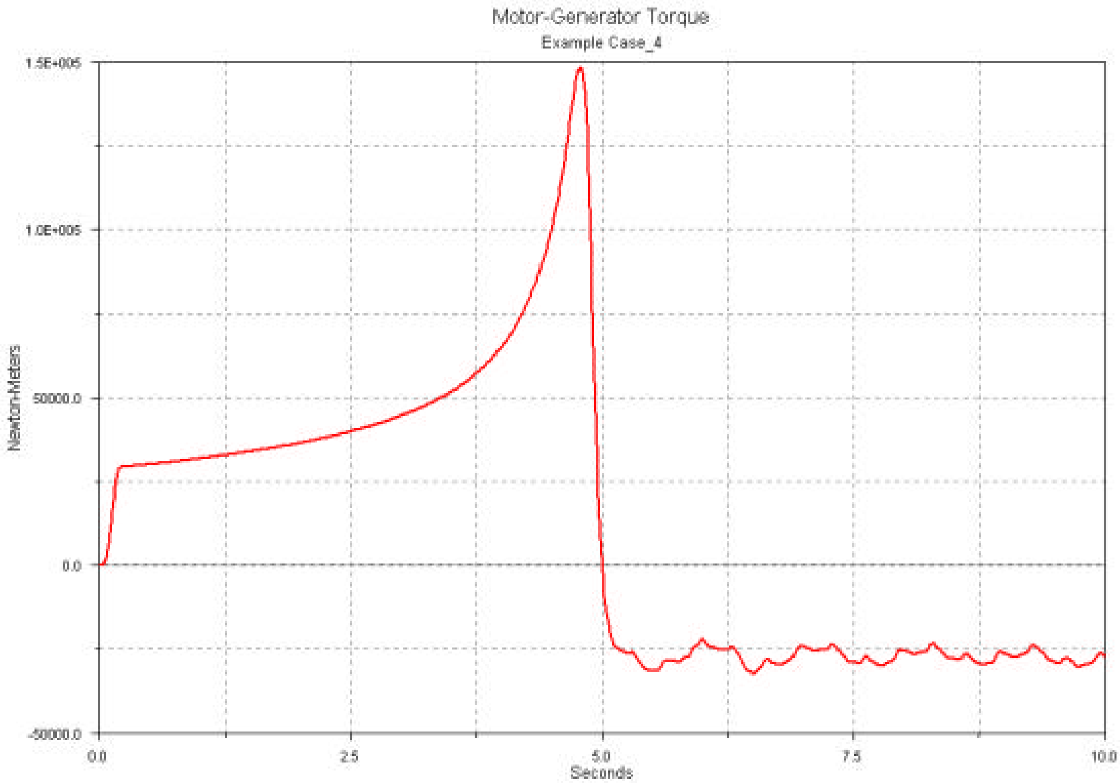
Yaw Table Loads (torques)
Example Case_4
80000.0
0.0
-80000.0
-1.6E+005
-2.4E+005
Newton-Meters
0.0
2.5
5.0
7.5
10.0
Seconds
X
Y
Z

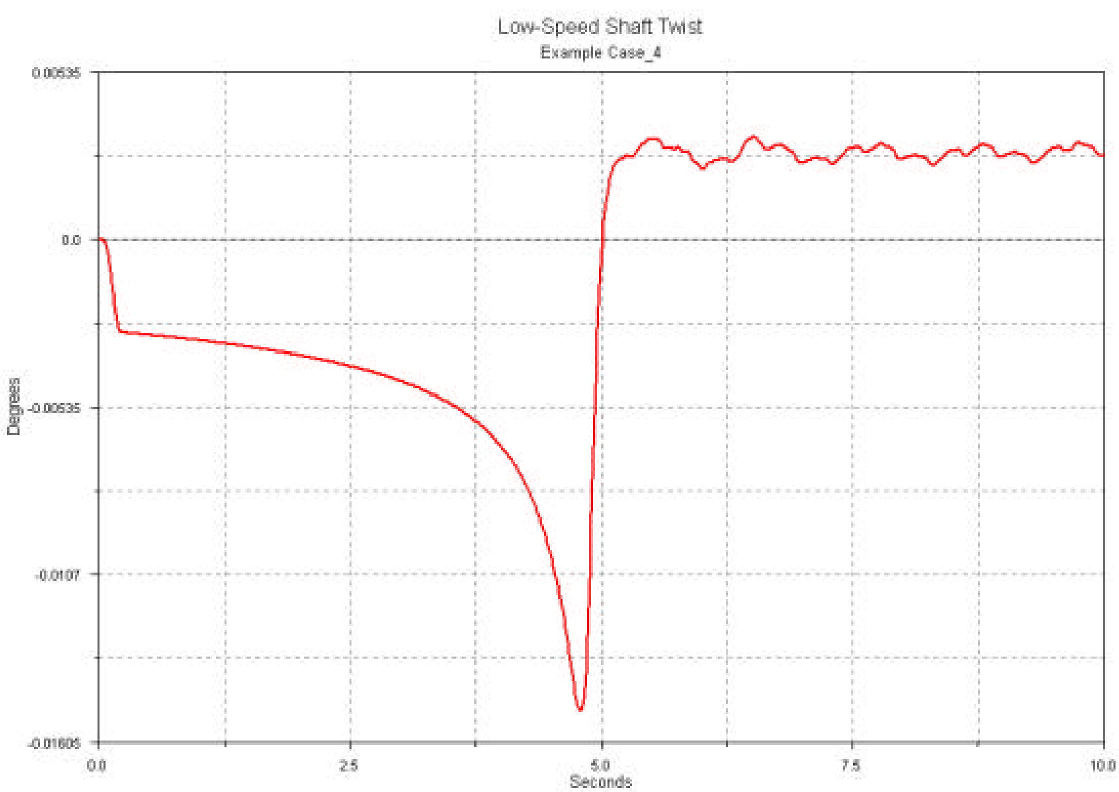

Rotor Speed
Example Case_4
61.5
41.0
20.5
0.0
-20.5
RPM
0.0
2.5
5.0
7.5
10.0
Seconds


Nacelle Yaw
Example Case_4
2.4
1.6
0.8
0.0
-0.8
Degrees
0.0
2.5
5.0
7.5
10.0
Seconds


Blade Root Loads (forces)
Example Case_4
1.005E+005
67000.0
33500.0
0.0
-33500.0
Newtons
0.0
2.5
5.0
7.5
10.0
Seconds
X
Y
Z


Blade Root Loads (torques)
Example Case_4
50000.0
25000.0
0.0
-25000.0
-50000.0
Newton-Meters
0.0
2.5
5.0
7.5
10.0
Seconds
X
Y
Z


Motor-Generator Torque
Example Case_4
1.5E+005
1.0E+005
50000.0
0.0
-50000.0
Newton-Meters
0.0
2.5
5.0
7.5
10.0
Seconds


Low-Speed Shaft Twist
Example Case_4
0.00535
0.0
-0.00535
-0.0107
-0.01605
Degrees
0.0
2.5
5.0
7.5
10.0
Seconds